N° d'ordre : 376-97 Année 1997

THESE

présentée

devant l'UNIVERSITE CLAUDE BERNARD - LYON I

pour l'obtention

du DIPLOME DE DOCTORAT

(arrêté du 30 Mars 1992)

par

M. RAKOTOMALALA Ricco

soutenue le 13 Décembre 1997

GRAPHES D'INDUCTION

JURY :	M. Kodratoff Yves	Rapporteur
	M. Matwin Stan	Rapporteur
	M. Bergadano Francesco	Examinateur
	M. Diday Edwin	Examinateur
	M. Duru Gérard	Examinateur
	M. Lamure Michel	Examinateur
	M. Zighed Djamel	Directeur

Remerciements

Les remerciements ont toujours un parfum de convenu qui peut paraître éculé et pourtant ils sont ô combien importants. En effet, un travail, quel qu'il soit, n'est jamais individuel, de nombreuses personnes ces dernières années m'ont aidé, guidé et finalement ont grandement contribué à la réussite de cette thèse.

Tout d'abord je pense au Pr. Zighed qui a su me persuader que j'etais capable de mener à bien cette tâche. Ces heures passées ensembles auront été pour moi autant de moments de "science intense" qui m'auront également permis de me faire un ami en qui j'accorde une grande confiance.

Je voudrais également remercier M. Chettouh qui m'aura permis, dans ma première année de thèse, de travailler sereinement. Son action a été salvatrice en me donnant les moyens de me consacrer pleinement à mon travail.

Je remercie M. Duru de m'avoir acueilli au sein de son école doctorale, j'ai beaucoup apprécié son inaltérable bonne humeur et son soutien sans réserves.

Je remercie MM. Kodratoff et Matwin d'avoir accepté d'être mes rapporteurs. Leurs remarques éclairées, les discussions impromptues que l'on a pu tenir, m'ont aidé à poser un autre regard sur mon propre travail. Leur disponibilité aura été sans faille, ce qui est appréciable dans ce monde où le temps est une denrée rare.

Je remercie MM. Duru, Lamure, Diday et Bergadano d'avoir bien voulu être mes examinateurs. Leurs suggestions tout comme leurs encouragements m'ont été précieux.

Un travail quel qu'il soit, ai-je dit, n'est jamais individuel, surtout en ce qui concerne la réalisation d'une thèse. Je tiens à adresser mes plus vifs remerciements à mes camarades de travail et amis : Valérie dont le sourire enjôleur aura fleurit le code source de DataManager, Serge, Marc, Sabine et Omar qui à un moment ou un autre ont partagé avec moi le bureau 12, et ont ainsi subit mes babillements d'apprenti chercheur, et puis tous les autres qui m'auront soutenu jusqu'à l'épreuve finale, lorsque mes jambes devenaient de plus en plus flageolantes.

Je voudrais enfin remercier les membres de ma famille. Leurs encouragements et leur confiance indéfectible ont été un appui incommensurable, une source de motivation qui m'a toujours soutenu tout le long de mes études. Je pense plus particulièrement à mes parents, Papa et Maman, qui de toute manière m'ont toujours aimé, Docteur ou pas... mais tant qu'à faire, Docteur c'est mieux.

Résumé

Les graphes d'induction en apprentissage ont fait l'objet de très nombreuses publications et ont atteint aujourd'hui un haut niveau de sophistication. Dans cette thèse, nous proposons une vision fédératrice de ces travaux sur les vingt dernières années. Certains d'entre eux reposent sur des arguments théoriques forts; d'autres, plus folkloriques, s'appuient sur le bon sens et une amélioration empirique des performances sur les bases de données exemples. En proposant un cadre commun d'étude de ces approches, ou du moins en essayant de situer le contexte de leur développement, nous constatons souvent qu'elles se rejoignent en cherchant à donner certaines qualités aux graphes construits. Notre objectif a été de produire un document de travail faisant l'état de l'art approfondi en délimitant au mieux les champs que nous couvrons (l'évaluation empirique des prédicteurs, les mesures d'évaluation des partitions, la détection de la taille optimale des graphes, l'extraction des règles, la généralisation des arbres de décision aux graphes d'induction, la construction itérative de variables de synthèse, l'agrégation de classifieurs) afin d'une part de situer les études présentes, d'autre part de définir, en connaissance de cause, les axes de recherche futurs. Notre principale conclusion est que les graphes construits de manière "classique", à partir d'un apprentissage unique sur un échantillon de la population originelle, sont proches de leurs limites en ce qui concerne le taux de reconnaissance en généralisation; la réduction de la taille du prédicteur en revanche connaît encore des améliorations significatives mais repose essentiellement sur des validations empiriques. Notre recherche a été le cadre du développement d'une plate-forme logicielle pour la comparaison de méthodes et l'extraction automatique de connaissances, SIPINA_W©, que nous diffusons dans le monde à travers l'Internet.

Table des matières

Table des figures

Liste des tableaux

Première partie

Généralités sur l'induction et leur évaluation empirique

Chapitre 1

Introduction

1.1 Apprentissage - Apprentissage supervisé

L'apprentissage automatique est certainement, en intelligence artificielle, le champ d'exploration le plus fertile de ces dernières années. Pourtant, pendant longtemps, on n'a pu donner une définition claire et précise du terme apprentissage. On sait de manière générale qu'une des prérogatives de l'intelligence est d'apprendre à partir de l'expérience passée pour adapter son comportement. L'apprentissage automatique est ainsi le champ d'étude où l'on essaie de mimer et de reproduire la capacité de l'homme à apprendre. [Simon, 1983] l'interprète comme les changements dans un système qui font qu'il accomplira mieux la même tâche, ou une tâche similaire, dans la même population dans l'avenir. [Dietterich, 1986] propose une approche plus fonctionnelle qui permet de l'évaluer, il le relie à la notion de "connaissances". Il distingue ainsi trois niveaux de description d'un système d'apprentissage : un système ne recevant aucune entrée et accomplissant mieux une tâche; un système qui reçoit des connaissances via des entrées mais n'accomplit aucune induction; et enfin, un système qui reçoit des entrées et en extrait des connaissances qui ne sont connues ni implicitement ni explicitement, c'est l'apprentissage inductif.

C'est cette dernière qui nous intéresse dans cette thèse, plus particulièrement l'apprentissage empirique qui vise à produire des règles générales à partir d'une série d'observations [Dietterich et Shavlik, 1990]. Formellement, nous caractériserons de la manière suivante l'inférence inductive. Soit D un domaine, composé d'une population Ω. Nous disposons d'un échantillon Ω^a et d'un algorithme d'apprentissage A. Sachant Ω^a et D, A produit une théorie M, issue de l'espace des hypothèses, que l'on peut utiliser pour expliquer la structure des données. Les objectifs peuvent être multiples : donner une description plus compacte des observations, distinguer les "structures" sous-jacentes qui régissent leur formation, prédire l'appartenance ou la valeur prise par un individu quelconque de la population originelle. L'inférence inductive recouvre deux domaines d'études pas nécessairement distincts selon le type d'information ingérée :

l'apprentissage non-supervisé et l'apprentissage supervisé.

Dans l'apprentissage non-supervisé, connu également sous le terme classification [Chandon et Pinson, 1981], l'algorithme A utilise un vecteur d'attributs $\overrightarrow{X} = (X_1(.), \ldots, X_p(.))$ pour essayer de trouver des "régularités" dans l'échantillon d'apprentissage. Elles se manifestent principalement par la constitution de groupes dans lesquels les observations diffèrent très peu au regard des valeurs prises par les $X_i(.)$, ces variables peuvent être continues ou qualitatives (prenant leur valeurs dans $X_i(\Omega) = \{x_{i1}, \ldots, x_{i\sigma_i}\}$).

L'apprentissage supervisé vise toujours à partir d'un vecteur d'attributs $\overrightarrow{X}$ que l'on nomme ici attributs prédictifs, ou encore variables exogènes, de reconstruire une fonction ou concept sous-jacent f telle que

$$Y = f(\overrightarrow{X})$$

$Y(.)$ est qualifiée de variable à prédire, ou encore de variable endogène. L'apprentissage permet de mettre à jour un modèle M, que l'on nomme classifieur ou prédicteur, tel que

$$\widehat{Y} = M(\overrightarrow{X})$$

avec pour objectif $\widehat{Y}(.) = Y(.)$.

Selon la nature de $Y(.)$, nous distinguons généralement deux familles d'apprentissage supervisé : lorsque $Y(.)$ est continu, on parle de régression; lorsqu'il prend ses valeurs dans un ensemble fini $\{y_1, \ldots, y_K\}$, l'espace des étiquettes ou encore les classes, on parle plutôt de classement, c'est le thème principal de notre thèse.

1.2 Finalités de l'apprentissage supervisé

Parmi les finalités de l'apprentissage supervisé figurent le diagnostic et la prévision. Prenons un exemple simple d'application : l'accord de crédit bancaire à un client. Le banquier aimerait savoir à l'avance si celui-ci est solvable. Il est évident qu'il le saura lorsque le client aura ou non remboursé son dédit, or justement la décision de l'accorder dépend de cette réponse. Le banquier aimerait alors quantifier le risque qu'il prend en accordant le crédit, cette décision dépend en fait de la variable à prédire "solvabilité" qui ne prend que deux modalités "solvable" et "non-solvable".

Généralement cette variable est difficile d'accès (le client n'avouera jamais qu'il ne remboursera pas) ou connue avec retard (lorsque l'on se rendra compte qu'il n'est pas solvable, il sera trop tard). L'objectif est, à partir de variables moins coûteuses et d'accès immédiat, d'essayer de prévoir la valeur prise par la variable "solvabilité" : ces variables peuvent être des informations signalétiques ou encore basées sur le comportement passé des individus. Dans notre exemple, si le client vient de sortir de prison pour escroquerie, qu'il est au chômage avec six enfants à charge, et que sa femme vient d'obtenir le divorce et une pension alimentaire conséquente, sans préjuger

de ses qualités intrinsèques, il apparaît peu probable que le client puisse rembourser quoi que ce soit.

Ce processus d'induction peut s'insérer dans des démarches plus générales d'extraction de connaissances ou de prévision. On distingue principalement deux champs d'application : l'extraction de connaissances à partir de données, l'alimentation de systèmes experts.

1.2.1 L'extraction de connaissances à partir de données (ECD)

Le développement du matériel informatique et la baisse des coûts ont permis à de nombreux organismes de constituer de grandes masses de données à moindre frais. On estime que la quantité de données dans le monde double tous les vingt mois [Kodratoff, 1997]. Malheureusement cette masse d'information, source potentielle de connaissances pour la compréhension de mécanismes sous-jacents régissant les phénomène apparents, reste trop souvent sous-exploitée. Par exemple, les données issues de la première étude du ciel au mont Palomar, achevée en 1960, ne sont pas encore complètement cataloguées.

Seule la mise en évidence des liens cachés ou des phénomènes de causalité non-triviaux éclaireront les décideurs lors des processus de choix. De fait, l'agencement des données et l'extraction des connaissances qu'elles recèlent sont devenues des enjeux stratégiques pour les organisations. Dans cette optique, on a vu émerger un nouveau domaine d'étude qui s'inscrit parfaitement dans ce créneau, il s'agit de l'extraction de connaissances à partir de données[1].

Une définition simple de ce concept serait [Fayyad *et al.*, 1996b] : "l'extraction de connaissances à partir de données est un processus non-trivial d'identification de structures inconnues, valides et potentiellement exploitables dans les bases de données". Il se situe à l'orée de plusieurs axes d'études comme l'intelligence artificielle, l'apprentissage automatique[2], les statistiques, la visualisation des données. Sa véritable spécificité tient surtout à la nature des données utilisées : les bases de données sont optimisées avant tout pour une exploitation efficace de la base en se référant à la théorie des systèmes d'informations.

L'ECD se réfère à une démarche complète d'exploitation des données intégrant les étapes suivantes :

1. la sélection des données et leur acquisition via les serveurs de bases de données;

2. le pré-traitement et/ou la transformation des attributs. En effet les méthodes de traitement de données exigent souvent une présentation particulière (nature des données, distribution conditionnelles et inconditionnelles...);

1. knowledge discovery in databases
2. machine learning

3. application d'un opérateur de fouille des données[3] qui met en évidence les formes sous-jacentes qui structurent les observations, et produit des modèles explicatifs ou prédictifs. Cette étape est primordiale, c'est le moment où l'on essaie de donner un sens aux phénomènes observés, elle implique le choix de la modélisation adéquate (hyperplans, fonctions de score, règles de production...) que l'on utilisera par la suite pour représenter ces nouvelles connaissances;

4. évaluation des formes extraites. L'objectif est de produire de la connaissance sur le domaine d'étude, il est important de s'assurer que les conclusions émises correspondent à des mécanismes réels. Cette évaluation est d'autant plus cruciale que l'on veut par la suite utiliser le prédicteur dans la population originelle;

5. enfin, dernière étape et non des moindres, conversion de ces connaissances en un produit opérationnel pour la prédiction, et/ou transcription de celle-ci sous une forme intelligible à l'homme.

Ce processus n'est pas rigide, il arrive souvent que l'on boucle à différents endroits car il est apparu que certaines spécifications se sont avérées insuffisantes. Par exemple, les structures mises à jour indiquent une corrélation certaine entre deux variables que l'on devrait recomposer (rapport poids-puissance pour les automobiles). On pourra alors ré-introduire cette nouvelle variable dans la construction du modèle. Ainsi, l'extraction de connaissance est éminemment interactive même si l'un des principaux objectifs est de traiter rapidement les grosses bases de données. Ce paradoxe qui n'en est pas un montre à quel point la phase préparatoire est importante dans l'acquisition des connaissances. En tous les cas, l'ECD est un domaine de travail en pleine expansion et les applications réelles ne manquent pas [Fayyad *et al.*, 1996a].

1.2.2 L'alimentation de systèmes experts

Les systèmes experts sont destinés à résoudre des problèmes naguère dévolus à des experts humains. Une de leur principales composantes est la base de connaissances, que l'on assimile souvent à une base de règles.

Traditionnellement, ces bases sont construites par un expert du domaine qui, à partir de son expérience et de ses connaissances, propose des règles de production. L'introduction de l'apprentissage supervisé, et notamment de l'induction de règles à partir d'exemples que nous introduirons plus loin, a permis de lever deux goulots d'étranglement dans la construction de ces systèmes :

1. l'accélération du processus d'acquisition des connaissances. [Clark *et al.*, 1994] ont noté que la construction pouvait aller de 180 années-hommes pour un système comportant 8000

3. data mining

règles (XCON, un configurateur de machines VAX) à 9 années-hommes pour le système BMT (configuration d'équipement pare-feux dans les immeubles) qui contient 30000 règles. Cette réduction se répercute également sur les coûts de maintenance.

2. une meilleure fiabilité face aux règles d'experts, et surtout une fiabilité que l'on peut quantifier. [Kononenko, 1993] relève dans plusieurs applications médicales que les règles issues de l'inférence surpassent en qualité de prédiction celles fournies par les experts.

Cela ne doit pourtant pas nous induire en erreur. Les experts humains resteront toujours des interlocuteurs privilégiés tant qu'ils pourront, mieux que n'importe quel algorithme, intégrer dans leurs constructions les connaissances du domaine. Cela se traduit généralement par un choix judicieux des attributs, de la forme du classifieur, et de sa complexité.

1.3 Qualités désirées d'un classifieur

Compte tenu des finalités et de l'utilisation des prédicteurs, il est clair que son évaluation diffère d'un domaine à un autre. On s'accorde pourtant à reconnaître l'importance de quelques critères globaux que nous explicitons dans cette section, certains ont été repris par [Craven et Schavlik, 1996].

1.3.1 La précision

C'est le critère certainement le plus important, en tous les cas le plus souvent cité, de l'apprentissage supervisé. Elle montre la capacité intrinsèque du classifieur à reconnaître la variable à prédire dans la population. Lorsqu'elle est totale, c'est à dire l'erreur est nulle, on peut penser à juste titre que l'on a trouvé une expression du concept à apprendre.

Notons qu'une précision parfaite sur le fichier d'apprentissage ne reflète pas nécessairement une bonne qualité de prédiction, nous discuterons de manière approfondie plus loin de la divergence entre la précision en apprentissage et la précision sur la population, nous devons surtout retenir que c'est le deuxième objectif qui nous préoccupe et que hélas nous ne pouvons pas l'observer directement.

1.3.2 La compréhensibilité

Dans la définition de l'ECD, l'exploitabilité de la connaissance en était un des principaux objectifs. Celle-ci passe par la compréhensibilité du modèle, [Michalski, 1983] argue que la connaissance extraite doit être sémantiquement et structurellement similaire à celles qu'un expert humain peut produire. Cette compréhensibilité est avantageuse pour plusieurs motifs :

1. validation : une connaissance ne peut être véritablement mise en oeuvre que si elle est

acceptée par ses utilisateurs, pour cette raison la plupart des systèmes experts comportent un module de description de la prise de décision. Le diagnostic par un expert permet de valider la pertinence d'une information;

2. découverte : l'exploration de données met souvent à jour des régularités que l'on ne soupçonne pas dans le domaine d'étude, soit parce que la formation de l'expert ne couvre pas cette partie de la connaissance, soit parce que temporellement de nouveaux phénomènes ont vu le jour. Quoiqu'il en soit, il est primordial pour le chercheur de savoir déchiffrer les connaissances extraites pour pouvoir en juger;

3. explication : prédire la classe d'un individu est une chose, expliquer pourquoi en est une autre. Un des reproches souvent adressés aux classifieurs "boîtes noires" est de présenter une solution sans que l'on sache comment il y est parvenu, de gros efforts sont d'ailleurs faits pour réduire l'opacité de certains modèles comme les réseaux de neurones [Andrews *et al.*, 1995] [Craven et Shavlik, 1994a];

4. analyse : la tâche d'apprentissage n'est jamais définitive, de nouvelles informations ainsi que des connaissances du domaine peuvent nous amener à améliorer manuellement le classifieur afin d'en augmenter les performances. Cette manipulation n'est possible que si nous pouvons appréhender les éléments qui la composent, et apprécier qualitativement les modifications introduites;

5. flexibilité : aucun algorithme ne détient la vérité absolue, sur un domaine donné certains marchent mieux que d'autres, plusieurs algorithmes peuvent également mettre à jour différentes facettes de la connaissance. Dans cette optique, l'agrégation des classifieurs d'origines différentes semble une voie assez intéressante [Wolpert, 1992]. Cela n'est possible que si les différents algorithmes proposent une même expression du modèle prédictif, ce qui est le cas par exemple des bases de règles qui sont d'autant plus intéressantes que l'expert lui-même peut y adjoindre de nouvelles connaissances sous la forme d'autres règles.

Ces différentes raisons éclairent chacune un aspect particulier de la compréhensibilité du modèle, mais finalement se rejoignent, surtout en ce qui concerne les graphes d'induction que nous présenterons en détail plus bas, dans la notion de complexité. Plus un modèle est complexe, moins on aura de facilité à le comprendre et inversement. Dans un cadre plus général, il est évident que l'on saisit mieux la partition de l'espace de représentation d'un perceptron simple que le découpage effectué par un perceptron multicouche, cette opacité augmente avec le nombre de couches cachées.

1.3.3 Autres critères

Ces deux premiers sont les plus souvent cités dans les publications, ils ne doivent pas nous en masquer d'autres qui sont tout aussi importants selon le domaine sur lequel on travaille.

Rapidité d'apprentissage

Même si les machines ont atteint un degré de perfectionnement inimaginable il y a une vingtaine d'années, les ressources ne sont pas inépuisables et il est toujours intéressant d'explorer des stratégies rapides de constitution de classifieurs, d'autant plus que la taille des bases de données exploitées sont souvent titanesques. Les algorithmes d'énumération ou de recherche exhaustive, même s'ils sont assez puissants d'un point de vue théorique, n'ont pas droit de cité dans ce contexte.

Mis à part la recherche d'algorithmes rapides [Catlett, 1991a], les autres pistes d'exploration sont le développement de l'utilisation de machines parallèles [Agrawal et Shafer, 1996] et la recherche de méthodes d'échantillonnage efficaces [Zaki *et al.*, 1996].

Rapidité de classement

Le problème de la rapidité se manifeste également dans la phase de généralisation. Dans certains cas, les services d'urgences d'un hôpital par exemple, il est vital que l'on puisse prendre une décision en un temps très court. [Tan et Schlimmer, 1990] ont défini la notion de complexité dynamique qui correspond au nombre de questions moyennes que l'on pose pour avoir une réponse. Dans la gestion de processus en temps réel, ce critère devient, au même titre que la précision, un des plus importants.

Rapidité et facilité de mise à jour

Un prédicteur n'est pas figé dans le temps tout simplement parce que les phénomènes ne le sont pas. Dans cette optique, il est très important d'avoir un algorithme qui puisse être facilement mis à jour i.e la recomposition du classifieur est très rapide, ou mieux encore, dont la mise à jour peut être faite de manière incrémentale, au fur et à mesure que les nouvelles observations arrivent. Dans le cadre des graphes d'induction, les principaux travaux sont l'oeuvre de [Utgoff, 1989a] [Utgoff, 1994].

Une très belle illustration de ce problème nous a été fournie par [Dietterich, 1996]. Dans la construction d'un contrôleur de disque dur, nous voulons construire un classifieur qui prédit à un instant donné quel sera le cylindre actif à l'instant suivant. Nous devons donc construire un classifieur qui puisse prédire à partir, par exemple, des 1000 dernières observations la prochaine zone d'accès. Il est clair que le classifieur ne peut être figé ici puisque le remplissage des cylindres

et des zones dépend essentiellement du degré de fragmentation et du taux de remplissage du disque dur. Périodiquement il est indispensable de reconstruire le classifieur.

1.4 Induction de règles à partir d'exemples - Les graphes d'induction

1.4.1 Induction de règles

Parmi les méthodes de classement, l'induction de règles à partir d'exemples tient une place particulière parce qu'elle est celle qui réalise le meilleur compromis entre performances en précision et compréhensibilité [Clark, 1990]. De plus, il s'agit certainement de la forme de connaissances la plus utilisée dans les systèmes experts [Mowforth, 1986].

Une règle, en logique propositionnelle d'ordre O^+, est de la forme

$$\text{Si Pr}\, \textit{émisse}\ \textit{Alors}\ \textit{Conclusion}$$

où Pr *émisse* est une conjonction de propositions du type attribut-valeur

$$X_i = Valeur$$

et conclusion

$$Y = y_k$$

Il existe de nombreuses méthodes qui produisent des classifieurs en utilisant explicitement cette forme dans l'induction. On distingue notamment la famille des algorithmes AQ [Michalski, 1983] parmi lesquels nous citerons AQ11 [Michalski et Larson, 1983], AQ15 [Michalski *et al.*, 1986], CN2 [Clark et Boswell, 1991]. Depuis la méthode originelle qui consistait à explorer par spécialisation une ou plusieurs règles entièrement consistantes pour chaque classe, de nombreuses améliorations ont été apportés : l'introduction de processus assurant une meilleure résistance au bruit, l'induction constructive enrichissant la formulation de la proposition.

Une autre famille d'algorithmes utilise un système de représentation très proche, il s'agit des listes de décision de [Rivest, 1987]. Dans ce cadre, l'ordre d'application des règles est fixé à l'avance. On remarquera d'ailleurs que le premier algorithme CN2 [Clark et Niblett, 1989] avait adopté ce type de représentation des connaissances. Le classifieur se présente de la manière suivante

$$\begin{array}{l} \text{Si Pr}\, \textit{émisse}_1\ \textit{Alors}\ \textit{Conclusion}_1 \\ \qquad \text{Si}\ \textit{non}\ \text{Si Pr}\, \textit{émisse}_2\ \textit{Alors}\ \textit{Conclusion}_2 \\ \qquad\qquad \text{Si}\ \textit{non}\ \ldots \end{array}$$

La dernière conclusion étant la désignation de la classe par défaut lorsqu'aucune prémisse n'a été activée.

Malgré les indéniables qualités des stratégies précédentes, nous avons préféré nous pencher sur une autre méthode , l'induction par graphes, qui présente l'avantage d'utiliser un système de représentation remplissant parfaitement les critères de précision et surtout de compréhensibilité, mieux peut-être que les règles puisqu'il existe de nombreux travaux pour essayer d'exprimer les bases de règles en arbres de décisions afin d'en améliorer la lisibilité [Michalski et Imam, 1994]. La transformation en règles pour alimenter les systèmes experts peut être faite sans pertes ni modifications de leurs propriétés en classement.

1.4.2 Graphes d'induction

La construction des graphes d'induction, plus connus sous le terme d'arbres de décision, est une discipline déjà ancienne. Les statisticiens en attribuent la paternité à [Morgan et Sonquist, 1963] qui les premiers ont construit des arbres de régression (la variable à prédire est continue), pour donner ensuite lieu à toute la famille des classifieurs AID [Sonquist *et al.*, 1971] [Gillo, 1972] [Kass, 1980], on considère généralement que cette approche connaît son apogée avec la monographie CART de [Breiman *et al.*, 1984]; en apprentissage automatique en revanche, il est d'usage de citer les travaux de [Hunt *et al.*, 1966] avec la méthode ACLS, les méthodes de référence suivant cette voie sont sans aucun doute les stratégies ID3 [Quinlan, 1979] et C4.5 [Quinlan, 1993a]. La notoriété de ces méthodes est telle qu'il paraît impensable de faire un article sur les arbres et graphes d'induction sans citer au moins une des méthodes CART, ID3 ou C4.5.

En France, dans les années 70, la méthode Elisée de [Bouroche et Tenenhaus, 1970] se place dans la mouvance statistique, les travaux de [Terrenoire, 1970] sur les pseudo-questionnaires sont plutôt à rapprocher de la théorie de l'information. On note surtout que de cette famille a émergé le concept de graphes d'induction qui se présente comme une généralisation des arbres [Picard, 1972] [Zighed, 1985].

Algorithme de base

La popularité de la méthode repose en grande partie sur sa simplicité désarmante. Il s'agit tout simplement de trouver un partitionnement des individus que l'on représente sous la forme d'un arbre tel que l'on minimise une mesure. Dans les méthodes génériques (CART, ID3...), on procède à une optimisation locale, sur chacun des noeuds on recherche la variable qui entraîne la meilleure segmentation de l'échantillon. Rendue ainsi très facile à programmer, l'implémentation des arbres sur ordinateur peut être faite à l'aide d'une simple procédure récursive.

Afin de mieux appréhender le fonctionnement de l'algorithme, nous allons prendre un exemple tiré de [Quinlan, 1993a]. Soit un fichier composé de 14 individus, avec 4 attributs prédictifs et

Numéro	Perspective	Température ($°F$)	Humidité (%)	Venteux?	Jouer
1	soleil	75	70	oui	oui
2	soleil	80	90	oui	non
3	soleil	85	85	non	non
4	soleil	72	95	non	non
5	soleil	69	70	non	oui
6	couvert	72	90	oui	oui
7	couvert	83	78	non	oui
8	couvert	64	65	oui	oui
9	couvert	81	75	non	oui
10	pluie	71	80	oui	non
11	pluie	65	70	oui	non
12	pluie	75	80	non	oui
13	pluie	68	80	non	oui
14	pluie	70	96	non	oui

TAB. 1.1 – *Un exemple de fichier d'apprentissage*

une variable à prédire *Jouer* prenant deux modalités $\{oui, non\}$(Table 1.1).

Ce fichier peut être représenté formellement par une matrice $O = (\overrightarrow{X}|Y)$ de données à n lignes et $p+1$ colonnes où

- $n = card(\Omega^a)$, avec Ω^a l'échantillon d'apprentissage à partir duquel on va construire le classifieur,
- p est le nombre d'attributs prédictifs.

Chaque individu $\omega \in \Omega^a$ est donc matérialisé par une ligne de O, avec les composantes suivantes $(X_1(\omega)|\ldots|X_p(\omega)|Y(\omega))$. L'objectif de la construction de l'arbre est de poser successivement différentes questions de manière à produire des groupes homogènes du point de vue de la variable à prédire. Chaque sommet du graphe est alors représenté par la distribution empirique des classes, voici la composition du sommet initial

	Effectifs
$Jouer = oui$	$card(\{\omega \in \Omega^a \,/\, Jouer(\omega) = oui\})$
$Jouer = non$	$card(\{\omega \in \Omega^a \,/\, Jouer(\omega) = non\})$

A partir de l'exemple de la table 1.1, nous avons construit un arbre de décision à l'aide de la méthode ID3 [Quinlan, 1979] : il en résulte le classifieur représenté dans la figure 1.1.

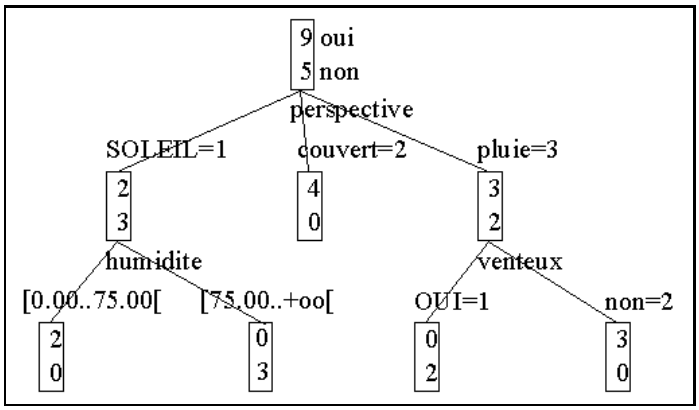

FIG. 1.1 – *Une partition de l'ensemble exemple*

Le détail du fonctionnement de l'algorithme peut être retracé à l'aide du listage des sous-groupes tour à tour constitués et de leur pureté du point de vue de la variable à prédire. A l'origine, l'échantillon d'apprentissage est composé de tous les individus observés $\Omega^a = \{1, \ldots, 14\}$.

La première question porte sur les valeurs prises par la variable *Perspective*, nous mettons ainsi à jour 3 groupes :

- $\Omega_{(Perspective=soleil)} = \{1, 2, 3, 4, 5\}$, avec une distribution des classes $\begin{array}{|c|}\hline 2 \\ \hline 3 \\ \hline \end{array}$;
- $\Omega_{(Perspective=couvert)} = \{6, 7, 8, 9\}$, avec la distribution de classes $\begin{array}{|c|}\hline 4 \\ \hline 0 \\ \hline \end{array}$, on peut penser que la partition peut être arrêtée sur ce sommet puisque l'on n'a que des représentants de la classe $(Jouer = oui)$;
- $\Omega_{(Perspective=pluie)} = \{10, 11, 12, 13, 14\}$, avec $\begin{array}{|c|}\hline 3 \\ \hline 2 \\ \hline \end{array}$

Si le second groupe donne satisfaction, il est pur et ne contient que des individus portant l'étiquette *oui* sur la variable *Jouer*, il en est autrement du premier et du troisième groupe sur lesquels nous reposons de nouveau des questions pour améliorer la discrimination. On décide d'utiliser la variable *humidité* en fixant comme seuil 75% sur le premier groupe :

- $\Omega_{(Perspective=soleil, Humidite<75\%)} = \{4, 5\}$ avec la distribution de classes $\begin{array}{|c|}\hline 2 \\ \hline 0 \\ \hline \end{array}$,
- $\Omega_{(Perspective=soleil, Humidite\geq 75\%)} = \{1, 2, 3\}$ avec la distribution de classes $\begin{array}{|c|}\hline 0 \\ \hline 3 \\ \hline \end{array}$

De la même manière, nous partitionnons l'ensemble $\Omega_{(Perspective=pluie)}$ mais en utilisant l'attribut *venteux* :

- $\Omega_{(Perspective=pluie, Venteux=oui)} = \{10, 11\}$ avec la distribution de classes $\begin{array}{|c|}\hline 0 \\ \hline 2 \\ \hline \end{array}$,
- $\Omega_{(Perspective=pluie, Venteux=non)} = \{12, 13, 14\}$ avec la distribution de classes $\begin{array}{|c|}\hline 3 \\ \hline 0 \\ \hline \end{array}$

Cette fois, nous avons réussi à isoler 5 groupes que l'on peut visualiser directement sur le graphe de la figure 1.1 avec les questions-réponses afférentes à leur construction.

Cette simplicité apparente ne doit toutefois pas cacher des problèmes réels qui se posent lors de la construction de l'arbre : le choix de la variable de segmentation, l'arrêt de l'expansion de l'arbre, l'assignation d'une feuille à une classe lorsque elles ne sont pas pures, ou encore le choix de la valeur seuil pour la segmentation sur variables continues.

Avantages de l'induction par graphes

La précision des graphes d'induction, malgré la simplicité du système de représentation des connaissances, est excellente face à des méthodes plus complexes [Quinlan, 1988b] [McKenzie et Low, 1992] [Quinlan, 1993b]. On peut avancer qu'au même titre que la formulation linéaire en régression, ils constituent certainement une approche de référence qui permet d'appréhender la complexité du concept et l'évaluation des outils les plus adaptés. Couplée avec sa lisibilité, on peut visualiser directement le processus de classement d'un nouvel individu, il est évident qu'ils représentent des instruments privilégiés d'exploration des données, que ce soit en termes de description ou de classement. Avec l'émergence de l'ECD, on voit d'ailleurs une recrudescence des travaux autour des graphes et des arbres de décision matérialisés par la diffusion de plus en plus large de logiciels spécialisés sous des plate-formes différentes. En effet si l'on se réfère aux critères que nous avons proposés dans la section 1.3, on constate qu'ils y répondent au mieux. Leur simplicité en fait un algorithme facile à programmer, qui requiert peu de calculs, particulièrement adapté pour le traitement de grosses bases de données. Sa mise à jour ne pose pas particulièrement de problèmes, il existe même des stratégies incrémentales qui, sous certaines conditions, retrouvent les graphes tels qu'ils seraient si l'on avait appris sur la totalité du fichier. Enfin, leur rapidité en classement dépend tout simplement du nombre moyen de noeuds en partant de la racine aux feuilles. Plus on marquera notre préférence à la simplicité, plus on satisfera à tous les critères d'évaluation des classifieurs mis à part la précision qui peut en souffrir dans certaines circonstances. En tous les cas, l'expert peut ici piloter à sa guise l'arbitrage entre la complexité et la précision.

En considérant d'autres critères, la liste des avantages des graphes en induction n'est pas close. On peut citer pêle-mêle la possibilité de construction ou de révision partielle du classifieur par un

expert [Kervahut et Potvin, 1996], la traduction exacte en base de règles d'un graphe en vue d'en améliorer les performances en classement et afin d'alimenter les systèmes experts [Quinlan, 1993a] [Rakotomalala, 1995a], la sélection des meilleurs attributs prédictifs, le traitement automatisé des données manquantes [Quinlan, 1989].... La discipline est en pleine maturité, son ancienneté plaide pour elle. Depuis les algorithmes de référence que sont CART [Breiman *et al.*, 1984] et ID3 [Quinlan, 1986b], les améliorations des nouvelles stratégies en terme de performances sont certes réelles mais finalement peu importantes au regard du surcoût qu'elles entraînent. Pour s'en persuader, il suffit de consulter la plupart des articles : tous utilisent des comparaisons avec l'un des deux algorithmes [4].

En fait, un de ses principaux attraits, peut-être le plus important ces dernières années, est le constat au fil des années que les graphes d'induction sont le théâtre d'innovations et de tentatives de formulations théoriques nouvelles en apprentissage. La liste est longue et il est difficile d'être exhaustif :

- l'introduction de la description minimale en apprentissage supervisé [Quinlan et Rivest, 1989] [Wallace et Patrick, 1993];
- l'introduction de la théorie bayesienne [Buntine, 1991] [Munteanu, 1996];
- l'évaluation directe d'un système d'apprentissage théorique en utilisant des arbres à complexité limitée [Auer *et al.*, 1995];
- l'étude approfondie de la discrétisation des attributs continus et de ses conséquences sur le classifieur [Zighed *et al.*, 1996] [de Merckt et Quinlan, 1996];
- la décomposition biais-variance de l'erreur et son traitement dans l'induction [Breiman *et al.*, 1984];
- l'opportunité du passage aux classifieurs agrégés [Breiman, 1996a] [Freund et Schapire, 1996];
- l'introduction et l'application directe de la théorie des ensembles flous [Janikow, 1993] [Ramdani, 1994] [Boyen et Wehenkel, 1996];
- le traitement de données non-tabulaires (i.e avec des variables pouvant prendre plusieurs valeurs pour un individu [Cohen, 1996]), ou plus généralement de nature imprécise [Ciampi *et al.*, 1995];

S'il est difficile ici de prétendre à l'exhaustivité, qu'en est-il de l'utilisation des graphes d'induction sur des problèmes réels ? Il faudrait une encyclopédie pour les recenser. Les graphes ont été appliqués avec succès dans de très nombreux domaines tels que la recherche médicale [Rabaseda, 1996] [Dowe *et al.*, 1992], la géographie [Lagacherie et Holmes, 1996], la prévention

4. souvent également avec C4.5 [Quinlan, 1993a] qui est un descendant direct d'ID3, et surtout dont le code programme est disponible sur Internet

des incendies [Dowe et Krusel, 1993], la surveillance des réseaux électriques [Wehenkel, 1995].... [Murthy, 1995] dans sa thèse en cite un grand nombre tirés des publications anglo-saxonnes parues entre 1993 et 1995.

1.5 Plan de la thèse et contributions

1.5.1 Motivations et plan de la thèse

Au cours de mon activité de recherche, à l'issue de nombreuses discussions dans des rencontres et congrès avec des chercheurs français en apprentissage automatique, il m'est apparu de plus en plus que malgré l'énorme succès anglo-saxon des graphes d'induction, ils restaient méconnus en France. L'idée générale qui a cours est que les méthodes à base de segmentation simple sont suffisamment puissantes, les autres axes de développement n'étant finalement que fioritures.

Nous ne partageons évidemment pas cette opinion. S'il est vrai qu'en terme de performances en généralisation, les progrès sont faibles ces vingt dernières années, ce qui semble naturel puisque l'on est de toute manière limité par le système de représentation de connaissances adopté, nous rejoignons complètement l'avis de [Buntine, 1991] selon lequel les graphes et arbres de décision sont devenus le champ d'innovations décisives en matière d'induction, on peut les considérer comme des méthodes "étalons" sur lesquelles on peut tester la viabilité de nouveaux paradigmes.

Notre propos dans cette thèse est de mettre en exergue les principaux axes d'innovations en matière d'induction par graphes en nous concentrant sur les points essentiels de leur construction. Notre objectif premier est d'essayer de donner une vision globale de l'évolution et des perspectives de la recherche dans cette discipline. Dans cette optique, nous avons divisé notre travail en quatre parties distinctes :

1. la première partie aborde les considérations générales autour de l'induction. Il est composé de ce chapitre introductif, et du deuxième chapitre consacré à l'évaluation et comparaisons de classifieurs. Les résultats et commentaires ne sont pas véritablement spécifiques aux graphes.

2. la seconde partie aborde l'algorithme de base de construction des arbres de décision, il est composé de quatre chapitres dont les intitulés sont dûs à la caractérisation de la méthode proposée par [Breiman *et al.*, 1984] :

   - le chapitre trois est consacré à l'étude de la sélection des variables de segmentation sur un noeud, notre principale contribution a été de proposer un essai de classification des mesures d'évaluation fondé sur leur nature et leur adéquation aux propriétés mis en exergue par [Zighed, 1985] dont nous discutons le sens et la pertinence;

- le chapitre quatre aborde une première fois la problématique de la recherche de la taille optimale en examinant les principales motivations autour de la préférence à la simplicité et de leur adéquation selon la nature du domaine étudié;
- le chapitre cinq s'attaque toujours à la détermination de la taille optimale mais en traduisant le problème de l'induction en un problème d'optimisation. Cette stratégie possède l'avantage d'être appropriée quelle que soit la structure du graphe adoptée. Nous introduisons dans ce chapitre une interprétation de la construction des graphes en terme de régression, l'induction serait alors assimilée à la recherche de la corrélation la plus significative entre la composition des attributs prédictifs et la variable à prédire. Enfin, puisque nous disposons de mesures globales permettant d'évaluer la partition, nous testerons également l'efficacité réelle de l'élagage face à la règle d'arrêt en terme de précision en classement du classifieur.
- enfin, le chapitre six, dernier de la deuxième partie, est consacré à l'extraction de règles dans les graphes d'induction. Nous y détaillons bien sûr le processus d'affectation classique d'une classe à une feuille, mais nous explorerons également d'autres pistes. La principale est la validation une à une des règles issues du graphe, travaillant à partir de l'intensité de l'implication de [Gras, 1979] [Lerman *et al.*, 1981], nous proposons un nouvel algorithme d'extraction de règles dans les graphes que nous présentons comme une alternative à l'élagage. Cette procédure entre autres nous permet de justifier la détermination de la taille minimale associée à une règle. Enfin, nous concluons ce chapitre en évaluant l'avantage que l'on peut retirer du changement de représentation que constitue le passage du graphe aux règles, notamment en ce qui concerne la simplification.

3. la troisième partie présente les "nouveaux" travaux pour améliorer les arbres de décision classiques. Nous ne faisons pas un tour d'horizon complet ici en nous contentant de citer les principaux résultats comme on peut le voir dans la plupart des travaux de survol. Nous préférons faire une sélection qui peut paraître arbitraire, mais qui nous permet d'étudier plus en profondeur les tenants et aboutissants de ces nouveaux paradigmes. Nous listons dans la section suivante les sujets qui ne seront pas adoptés dans cette thèse.

   - le chapitre sept est consacré aux graphes proprement dit. Le passage des arbres aux graphes n'est pas une généralisation qui s'appuie essentiellement sur une coquetterie terminologique, il répond à des problèmes réels qui tiennent aux insuffisances du système de représentation par les arbres. Nous y développerons les différentes manières de construire les graphes et nous évaluerons leur efficacité sur des bases réelles;
   - le chapitre huit est une autre manière de contourner les faiblesses des arbres en aug-

mentant cette fois la puissance de représentation des attributs via des variables intermédiaires. Nous y esquisserons la problématique de constructions de variables de haut niveau dans le cadre des données continues, nous testerons alors un essai utilisant l'analyse en composantes principales;

- toujours dans la problématique de l'adaptation des variables à l'induction par graphes, dans le chapitre neuf, nous étudierons avec soin le découpage en intervalles d'une variable continue. Ce domaine d'étude est certainement celui qui a le plus bénéficié de l'impulsion de la communauté des chercheurs sur les graphes d'induction, la plupart des travaux marquants sont l'oeuvre d'auteurs qui ont travaillé sur les graphes. Dans ce chapitre, nous présentons une méthode d'exploration optimale de complexité réduite [Zighed *et al.*, 1997] inspirée des travaux de [Lechevallier, 1990] et [Fayyad et Irani, 1992], nous étudierons ses implications sur les performances en induction.
- le chapitre dix est consacré à l'étude de l'agrégation de classifieurs et de leurs justifications dans le cadre de la construction des graphes d'induction. Nous constaterons surtout que l'on peut en dériver de nouvelles stratégies pour construire des graphes plus puissants et précis en généralisation.

4. la quatrième partie est consacrée aux applications et aux perspectives.

   - dans le chapitre onze, nous présentons une plate-forme d'ingénierie des connaissances que nous diffusons à partir du serveur WEB de notre laboratoire. Ce logiciel reprend en grande partie la problématique de l'extraction automatique de connaissances à partir de données;
   - dans le chapitre douze, nous décrivons les principales études qui ont été menées à l'aide de ce logiciel, plus généralement nous essayerons de discerner les différents contextes de son utilisation;
   - enfin, nous conclurons dans le treizième et dernier chapitre en essayant de dégager les points essentiels mis en exergue dans cette thèse, nous y discuterons de l'opportunité et du devenir de la recherche sur les graphes d'induction.

Nous étions assez partagés quand au style de présentation de cette thèse. Nous aurions pu adopter une démarche plus classique de "Contributions à ..." en mettant en avant les résultats de nos travaux sur l'extraction de formes par évaluation et validation statistique des règles, l'interprétation en terme de régression de l'induction en utilisant le schéma d'analyse de variance sur données catégorielles, ou encore la discrétisation optimale des attributs continus. Mais finalement, nous avons préféré replacer entièrement ces travaux dans leur contexte, et les présenter

plutôt comme des alternatives s'appuyant sur des formulations différentes de problèmes déjà connus, peut-être au risque de les noyer dans les différents chapitres. En tous les cas, cela nous a permis d'adopter une démarche unifiée de description générale de l'induction par graphes.

1.5.2 Les domaines non-abordés dans cette thèse

La plupart des thèses que nous avons consultées comportent une partie "survol" qui s'attache à citer les principaux travaux ces dernières années. Le plus admirable est certainement le travail de [Murthy, 1995] qui nous a servi à bien des égards de point de départ. Hélas, dans celui-ci comme les autres, les auteurs se contentent généralement de citer les travaux avec, pas toujours, une courte description. Nous avons voulu éviter cet écueil en proposant un survol un peu plus réduit mais discutant de manière approfondie certains points qui semblent sujets à caution. Notre objectif est d'étudier les graphes d'induction en tant qu'outil d'apprentissage automatique d'un classifieur de performances acceptables et bénéficiant d'une grande lisibilité. Nous avons ainsi mis de côté des sujets qui auraient pu constituer un chapitre de notre thèse mais dont nous avons différé l'étude parce qu'ils s'écartent sensiblement de notre objectif initial. Fort heureusement, un travail de recherche est un sacerdoce qui n'est jamais terminé, ces sujets feront certainement l'objet d'études poussées de notre part dans un autre cadre que la thèse de doctorat :

- les arbres flous : en conformité avec la théorie des ensembles flous [Zadeh, 1965], il a été développé ces dernières années plusieurs systèmes de construction d'arbres intégrant notamment la notion de coupure floue déjà testée dans un cadre tout à fait différent par [Carter et Catlett, 1987]. [Dietterich et Kong, 1995b] affirme que l'amélioration des performances repose surtout sur la réduction de la variance, les segmentations sont moins sensibles aux individus bruités, ou se situant au alentours de la borne de discrétisation. [Marsala, 1994] a montré que rien qu'en "fuzzifiant" des bornes déterminées de manière classique, on pouvait sur le fichier d'ondes de [Breiman *et al.*, 1984], réduire le taux d'erreur en généralisation. Si la lisibilité de l'arbre en tant que classifieur est partiellement préservée, il est évident ici que la traduction en règle pour les systèmes experts pose problème dans la description des segments de coupure floue.

- les arbres sur données symboliques : les données peuvent être entachées d'incertitudes, ou même ne pas être mono-valuées. L'étude des données symboliques est un des axes d'études privilégié de [Diday, 1995], l'application dans les graphes est l'oeuvre de [Ciampi *et al.*, 1995]. L'objectif ici n'est pas tellement de réduire l'erreur mais plutôt d'adapter l'algorithme initial à des données de natures différentes, plus riches que les données classiques.

- la construction interactive : un graphe d'induction décrit un processus de prise de décision. Dans certain cas il peut être intéressant de laisser à l'expert le soin de construire le clas-

sifieur en posant les questions adéquates, une telle action est possible parce que l'on peut constater de visu les conséquences de chaque opération dans le graphe. De fait, il peut être intéressant parce que moins coûteux d'introduire certaines variables en premier et d'en exclure d'autres. Avec la construction interactive, on entre de plain pied dans l'assimilation des connaissances du domaine dans la construction du classifieur. En France, les travaux, surtout dans le domaine de la médecine, ont été entre autres l'oeuvre de [Cremillieux, 1991] [Cremillieux et Robert, 1996].

– les arbres obliques : les graphes classiques, par la discrétisation statique des attributs continus induisent des découpages de l'espace de représentation que l'on nomme couramment de "parallèles aux axes". [Breiman *et al.*, 1984] ont été parmi les premiers à évoquer la possibilité d'introduire des découpages plus riches, constitués d'hyperplans séparateurs construits par combinaison linéaire de variables. Ces approches ont donné lieu ultérieurement à de nombreux développements [Utgoff, 1989b] [Murthy *et al.*, 1994] [D'Alche-Buc *et al.*, 1994]. Si l'efficacité de ces méthodes ne souffrent pas de contestation, on peut néanmoins se poser des questions sur la lisibilité du prédicteur associé, une combinaison linéaire mettant en jeu plusieurs variables est assurément plus difficile à interpréter qu'un découpage simple sur une variable. Le gain en précision ici est en balance avec l'accroissement de complexité (au moins en description).

– l'optimisation globale : la recherche du l'arbre le moins complexe consistant sur les données est NP-complet [Takenaga et Yajima, 1993], il en est de même si l'on recherche la configuration minimisant un indicateur global de qualité de partitions. La stratégie de base des graphes d'induction est gloutonne, [Murthy et Salzberg, 1995a] ont montré expérimentalement la bonne tenue d'un tel algorithme. Mais dans la mesure où l'on conjecture une forte corrélation entre l'indicateur de qualité de la partition et la précision en généralisation, on peut se demander légitimement si l'optimisation du premier n'entraîne pas une amélioration du second. Dans cette optique, de nombreuses méthodes d'optimisation ont été testées dans les arbres, sans que vraiment leur introduction ait été décisive [Koza, 1991] [Weiss et Indurkhya, 1993]. En revanche le surcoût en terme de complexité de calcul est réel.

– la construction incrémentale : surtout lorsque les données sont très nombreuses, il est assurément plus avantageux de remettre à jour un classifieur existant plutôt que de tout réapprendre. [Utgoff, 1989a] a été un des promoteurs de l'approche incrémentale, il a amélioré plus tard son algorithme de manière à retrouver les mêmes résultats qu'ID3 appliqué sur la totalité des données [Utgoff, 1994]. Il reste cependant, à l'instar de notre exemple plus haut, que ce domaine d'étude est très spécifique.

Bien que nous n'aborderons pas explicitement ces domaines, nous y ferons bien sûr mention lorsque les circonstances, ou une meilleure explicitation des différentes problématiques, l'exigeront.

Chapitre 2

Evaluations et comparaisons empiriques de classifieurs

2.1 Introduction

Parmi les objectifs fondamentaux de l'apprentissage automatique figure la volonté de construire de "meilleurs" classifieurs. L'adjectif "meilleur" répond souvent, mais pas toujours, au taux d'erreur en généralisation. Dès lors il importe de pouvoir qualifier les performances des nouveaux algorithmes d'autant plus que beaucoup d'entre eux consistent en des modifications mineures, mais considérées comme essentielles par ses promoteurs, de stratégies déjà existantes.

La première méthode est essentiellement verbale et/ou mathématique : on montre une série de problèmes sur lesquels les méthodes existantes échouent parce qu'elles ne sont pas adaptées, on propose alors un nouvel algorithme qui répond de manière adéquate à ce type d'écueil. On note d'ailleurs que la plupart des papiers en apprentissage suivent à la lettre ce plan et nous n'y dérogeons pas dans certains de nos chapitres. Cette première étape est très importante car elle justifie l'introduction de nouvelles solutions, en revanche elle n'indique pas son comportement réel lorsqu'il est mis à l'épreuve i.e l'apprentissage sur des données. C'est le rôle de l'évaluation expérimentale.

Dans une enquête relativement récente, [Prechelt, 1996] a constaté que dans le domaine des réseaux de neurones par exemple, très peu de papiers testaient leur stratégies sur des données réelles. Sur un échantillon de 190 articles parus dans quatre journaux[5] en 1994, seulement 8% d'entre eux ont utilisé au moins une base réelle, 29% n'ont inclus aucune évaluation empirique. Fort heureusement, cette situation est appelée à changer avec l'apparition de serveurs de données tests[6] qui présentent le double avantage d'être parfaitement documentés et d'avoir fait l'objet

5. Neural networks, Neural Computation, Neurocomputing, IEEE Transactions on Neural Networks
6. les fichiers benchmarks

d'autres études par ailleurs, ce qui permet de mieux connaître leurs caractéristiques réelles et les performances que l'on peut attendre de telle ou telle famille d'algorithme. De nos jours, il semble impensable de publier un papier où l'on introduit un nouvel algorithme sans procéder à des tests. Bien entendu, dans les autres cas où l'on procède à des études théoriques, ces tests sont complètement superfétatoires.

Le champ couvert par l'étude expérimentale est très vaste, elle semble particulièrement adaptée à l'apprentissage automatique. Tout d'abord, elle permet de vérifier la viabilité des solutions proposées en classement. Le déroulement manuel de l'algorithme sur des petites bases reste une des manières essentielles de montrer la bonne tenue et le fonctionnement d'un algorithme. De plus, si l'on se réfère au taux d'erreur, la borne inférieure de performances est le choix de la classe la plus fréquente dans l'échantillon d'apprentissage. Si le nouvel algorithme propose des taux d'erreurs supérieurs sur de nombreuses études, il est clair qu'il y a un sérieux problème et que l'on doit reconsidérer la proposition.

En second lieu vient le corollaire du premier argument : l'analyse de la fiabilité de l'algorithme. Cela inclut bien sûr une estimation honnête du taux d'erreur commis en généralisation qui permet de quantifier le risque encouru en adoptant le classifieur, mais également l'analyse du type d'erreur commis (sur quelles classes la confusion est-elle la plus manifeste? les coûts sont-ils symétriques?) et la caractérisation des situations de leur occurrence (niveau de bruit, "forme" du concept, taille de l'échantillon d'apprentissage). En ce sens, la décomposition biais-variance de l'erreur en classement introduite par [Breiman *et al.*, 1984] [Breiman, 1994] constitue une avancée formidable qui a permis l'initiation de nouvelles pistes de recherche et justifier a posteriori d'autres solutions déjà anciennes [Dietterich et Kong, 1995b].

En troisième lieu figure la sélection des modèles. N'oublions pas que notre objectif initial est de trouver le meilleur classifieur. Partant de critères fixés à l'avance, qui peuvent être différents selon les domaines d'étude, nous en discuterons plus loin, peut-on trouver une méthode qui soit la mieux adaptée? La sélection des modèles est un des champs les plus lucratifs de l'analyse expérimentale [Schaffer, 1993b] [Prechelt, 1996], elle répond objectivement à la question "quelle est la meilleure méthode sur mes données compte tenu des critères que l'on s'est fixé?". On peut ainsi comparer des stratégies aussi diverses que différentes que les approches connexionistes et les approches symboliques [Shavlik *et al.*, 1991]. Dans la plupart des cas, les auteurs travaillent sur le taux d'erreur parce que c'est le seul indicateur qui soit véritablement comparable d'un algorithme à un autre, mais on peut également envisager d'autres critères comme la complexité, le temps de réponse et la difficulté de mise à jour.

Enfin, quatrième et dernier argument en faveur de l'étude expérimentale, peut-être le plus ambitieux, est la réponse à la question "l'algorithme A est-il en général meilleur que B?". C'est un vaste problème et de nombreux papiers ont été publiés sous ce label, en s'appuyant d'ailleurs très souvent sur des expérimentations à plus ou moins grande échelle. Il est clair que le problème

est délicat, [Schaffer, 1994] et [Wolpert, 1996] ont démontré que l'avantage d'une méthode sur une autre dans un domaine donné sera sûrement contrebalancé par des performances moindres dans d'autres domaines. Leur analyse reposait néanmoins sur une hypothèse très forte d'équiprobabilité des concepts, ce qui est, on le sait, faux puisque dans ce cas il ne sert à rien d'apprendre, il n'y a pas de régularités dans les données. Il reste néanmoins que l'on doit appréhender avec circonspection cette question, de nombreux auteurs pensent à juste titre que les études multipliées sur des domaines très variés permettent d'y répondre certainement d'une meilleure manière que les expérimentations sur une seule base de données. Encore faut-il avoir l'assurance que les bases utilisées couvrent véritablement une vaste classe de problèmes.

Après le temps de disette est venu le temps de l'exagération. Même si l'on peut considérer que le développement des études empiriques est sans conteste un indicateur de maturité de la discipline [Salzberg, 1997], on a trop souvent vu apparaître des publications frappées du syndrôme de l'empirisme. Ainsi, de nombreux papiers ont été publiés sous le prétexte que leur méthode permettait de gagner quelques pourcents de plus sur des bases de données réelles, pour la plupart issu du serveur UCI Irvine [Murphy et Aha, 1995], d'autres encore consistent tout simplement à comparer empiriquement le comportement de logiciels déjà connus. On voit ainsi souvent apparaître les grands tableaux de résultats sur lesquels d'étranges ratios sont exhibés pour persuader le lecteur de la supériorité d'un algorithme face à un standard de la famille des classifieurs considérés, représenté souvent par la méthode C4.5 de [Quinlan, 1993a].

Notre objectif dans ce chapitre est de clarifier les conditions d'applications des études empiriques, et les conclusions que l'on peut en tirer. Nous aborderons ainsi dans un premier temps la caractérisation des bases de données. Puis nous discuterons des principaux critères d'évaluation des graphes d'induction. Nous détaillerons alors les méthodes d'évaluation et de comparaison d'algorithmes, dans la littérature on ne considère souvent que le taux d'erreur, nous verrons que l'on peut les élargir sans problèmes à d'autres critères. Nous conclurons enfin.

2.2 Les critères à étudier

Selon les qualités d'un classifieur exprimées dans le premier chapitre, on distingue généralement les critères suivants pour évaluer un graphe d'induction :

1. le taux d'erreur en généralisation, c'est bien entendu le critère roi de l'apprentissage automatique, il indique la capacité de l'algorithme à reproduire le concept à apprendre. Si pendant longtemps on a cherché à le réduire jusqu'à zéro, on s'est rendu compte peu à peu que dans certaines classes de problèmes comme la prédiction de la structure secondaire d'une protéine [Lerman et Costa, 1996], il y avait un seuil limite en-deçà duquel il est impossible de descendre. On considère généralement qu'il s'agit des performances du

classifieur bayésien. Remarquons que cet indicateur se prête autant à l'estimation qu'à la comparaison entre modèles.

2. le second critère est la complexité que l'on peut traduire à l'aide de plusieurs indicateurs : [Quinlan, 1987a] préfère le nombre de noeuds dans le graphe, en fait nous pourrions tout aussi bien prendre le nombre de feuilles. Pour un arbre binaire, la corrélation est parfaite, dans le cas contraire elle dépend du nombre de modalités des variables introduites dans le classifieur et de la présence en grand nombre ou non d'attributs continus. Mis à part les cas où l'on effectue un retraitement des connaissances lors du passage d'un arbre à une base de règle[7], on constate généralement que le nombre de feuilles correspond exactement au nombre de règles, et les noeuds au nombre de propositions. La notion de complexité qui caractérise en premier lieu à la lisibilité du modèle, répond également à d'autres exigences qui se rapprochent des performances en classement, il s'agit de la fiabilité du modèle : si le nombre de feuilles du graphe est faible, cela implique que les hyperrectangles induits contiennent en moyenne un plus grand nombre d'individus. Ainsi les décisions prises à partir des distributions de classes seront plus stables. [Cheng *et al.*, 1988] et [Goodman et Smyth, 1988] l'interprètent comme un critère de généralité des règles. Dans nos tests, nous utiliserons le nombre de noeuds dans le graphe, y compris les feuilles. Notre principale motivation a été le constat d'une plus grande variance par rapport au nombre de feuilles, ce qui nous semble de bonne augure pour comparer les différents classifieurs. Contrairement au taux d'erreur précédent, ce critère ne prend véritablement son sens que dans la comparaison, il semble peu utile dans l'absolu de savoir qu'un classifieur a produit quatre règles, en revanche cette information prend tout son sens si l'on apprend qu'avec une autre méthode nous obtenons 10 règles.

Dans la plupart des études empiriques que nous avons consultés, une estimation de ces deux indicateurs ou de leurs variantes est systématiquement mentionnée. Il reste néanmoins qu'il ne s'agit pas là de critères universels, tout dépend des préoccupations de l'expert. [Catlett, 1991a] par exemple, qui axe son étude sur les grosses bases de données, avance le critère de rapidité d'apprentissage. Nous pouvons également mettre en avant le temps de réponse lorsqu'une question est posée.

2.3 Les données tests

Il n'y a pas de type des données privilégiées mises à part celles du domaine sur lequel on travaille. Dans l'analyse de la boîterie, [Rabaseda, 1996] a travaillé spécifiquement sur des données

7. c'est le cas par exemple si l'on effectue à des validations ou des simplifications, nous reviendrons en détail sur ce problème dans le chapitre sur les règles

issues d'appareils de mesures quantifiant le déplacement du bassin en trois dimensions. L'objectif bien évidemment était de trouver la méthode et les paramètres adéquats sur ce domaine.

Mais l'étude expérimentale peut avoir d'autres motivations comme la caractérisation des algorithmes, ou encore l'analyse de leur comportement. De ce point de vue, nous distinguons trois types de données que nous présentons ici, nous expliciterons leurs relations avec les objectifs de l'expérimentation.

2.3.1 Données synthétiques

Ce sont des données artificielles générées soit à partir de fonctions booléennes ou arithmétiques. Elles présentent l'immense avantage d'être parfaitement maîtrisées, le concept à apprendre est connu d'avance. Elles servent généralement pour la mise au point. Produites en petites quantités, on peut les utiliser pour expliquer le fonctionnement d'un algorithme en le faisant "tourner à la main". Elles peuvent également servir pour caractériser les situations dans lesquelles une méthode marche mieux qu'une autre, ainsi [Pagallo et Haussler, 1990] se sont appuyés sur des exemples très simples de données synthétiques pour montrer l'incapacité des arbres à traduire simplement les formes booléennes disjonctives.

2.3.2 Données réalistes

Ce sont également des données artificielles mais générées avec un modèle proche de problèmes réels. [Prechelt, 1996] en distingue trois variantes :

- les données représentant des modèles mathématiques ou physiques complexes,
- les données générées par un modèle mathématique chaotique,
- les données simulant un processus stochastique en utilisant des fonctions de répartitions multidimensionelles.

Par rapport aux données réelles que nous présentons plus bas, elle possèdent le grand avantage d'être maîtrisées : on en connaît généralement le taux d'erreur du prédicteur bayésien, on en connaît également les principales caractéristiques (difficultés, complexité...) et on peut en modifier les propriétés de manière contrôlée (bruit, données manquantes...). De plus, on peut générer autant d'observations que l'on veut.

Ce type de données sont couramment utilisés pour estimer les performances des algorithmes sur des tâches similaires, elles peuvent ainsi permettre de délimiter le champ privilégié d'action de telle ou telle famille de classifieurs. Les fameux fichiers de caractères numériques et d'ondes de [Breiman *et al.*, 1984] en font partie.

2.3.3 Données réelles

Ce sont des données à part parce qu'elles proviennent d'observations réelles et constituent véritablement le baptême de feu des algorithmes qui seront appelés à traiter de problèmes concrets. Leur principal inconvénient est qu'elles sont complètement incontrôlées, on n'en connaît à l'avance ni le niveau de bruit, ni les relations des observations manquantes avec le concept à apprendre (il arrive souvent que les non réponses correspondent à une situation particulière), ni bien sûr le concept à apprendre (s'il existe).

Caractérisation des données réelles

Afin de bien appréhender ce type de données, il est nécessaire d'en circonscrire les principales caractéristiques numériques, et les informations afférentes. Généralement, on essaie de répondre au mieux aux questions suivantes avant de procéder à une analyse, certaines ont été reprises chez [Catlett, 1991a] et [Chandon et Pinson, 1981] :

- quel est l'effectif total?
- combien de classes comporte la variable à prédire?
- quels en sont les distributions, conditionnelles et inconditionnelles?
- combien y a-t-il d'attributs?
- sont-ils tous qualitatifs, ou continus, ou encore mixtes?
- quelles sont les distributions de ces attributs?
- y a-t-il des valeurs manquantes?
- quel est niveau de bruit sur les observations? Dans un premier temps, avant toute phase d'apprentissage, cela revient surtout à une étude univariée du bruit (erreur de mesure, perturbations...)

En répertoriant ces renseignements, on peut déjà se donner une idée de l'adaptation de telle ou telle famille de classifieurs. Par exemple, si tous les attributs sont continus, il est évident que les meilleures performances ne seront pas atteintes à l'aide des graphes d'induction sauf cas rarissime où les frontières entre les classes seraient parallèles aux axes de projection. En fait, si l'on se cantonne aux graphes d'induction, rien qu'en répondant à ces questions on se donne une idée de la stratégie à adopter. Nous verrons notamment dans le chapitre consacré à la détermination de la taille optimale des graphes que de nombreuses études empiriques sur données synthétiques, ironiquement, ont donné de nombreuses indications sur les meilleures pistes lorsque les données réelles présentent des caractéristiques particulières.

La question des serveurs de bases de données

Depuis quelques années, on a vu se développer des serveurs de données, où des bases ayant servi à d'autres études sont soigneusement répertoriées. Le plus célèbre dans la communauté de l'apprentissage automatique est sans aucun doute le serveur F.T.P. de l'Université UCI Irvine en Californie [Murphy et Aha, 1995]. De nombreuses études empiriques sont basées sur les résultats obtenus sur les fichiers qui y sont disponibles, avec l'hypothèse sous-jacente qu'ils représentent une vaste classe de problèmes réels. Il semblerait cependant que cette assertion est erronée. [Holte, 1993] entre autres s'est aperçu qu'ils correspondent en réalité à des concepts simples, et sur la plupart de ces fichiers on a intérêt à opter pour une forte préférence à la simplicité du modèle à apprendre.

On peut penser effectivement que les "donateurs" de ces fichiers les ont déposés parce que leur stratégie d'apprentissage s'y adaptait bien, on sait également qu'une large catégorie de données qualifiées de sensibles (données militaires...) ne seront jamais disponibles au grand public. Doit-on pour autant condamner ces serveurs?

Nous pensons qu'il faut prendre avec circonspection les résultats issus de ces traitements. Finalement les dangers sont d'une part la généralisation intempestive des résultats à des catégories de problèmes qui n'y sont pas représentés, d'autre part l'acharnement à trouver les meilleurs performances sur ces fichiers. Si l'on tient compte de ces restrictions, il est clair que ces bases sont de formidables outils à la disposition des chercheurs aux fins de tests et de comparaisons. Pour notre part, nous avons sélectionné aléatoirement 15 bases du serveur sus-cité dont les caractéristiques numériques sont consignées en annexe. Il est évident que nos commentaires tiendront compte, autant que possible, de la spécificité des fichiers utilisés.

2.4 Analyse et estimation de l'erreur

Le taux d'erreur en généralisation disions-nous était un critère roi en apprentissage automatique. Cela est indéniable, mais d'un autre côté il nous semble au moins aussi important d'analyser la structure de l'erreur.

2.4.1 La matrice de confusion

Elle se présente sous la forme d'un tableau de contingence confrontant la classe décidée (en colonne) avec la classe observée (en ligne) des individus composant l'échantillon. Nous disposons de deux types d'informations :

- le nombre de fois où le modèle s'est trompé,
- l'erreur commise lors du classement.

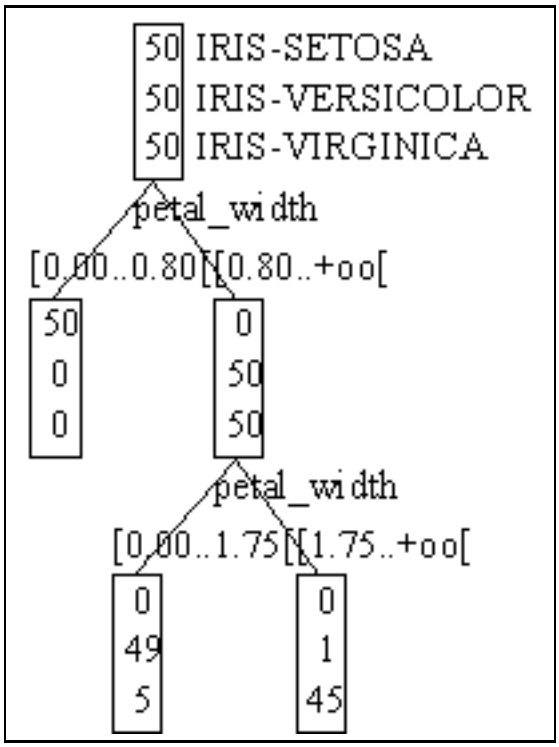

FIG. 2.1 – *Un exemple de traitement par SIPINA sur le fichier "Iris"*

A partir du graphe construit sur le fichier des Iris (figure 2.1) où la variable à prédire prend trois valeurs $\{Setosa, Versicolor, Virginica\}$, nous obtenons la matrice suivante :

	Setosa	Versicolor	Virginica
Setosa	50	0	0
Versicolor	0	49	1
Virginica	0	5	45

La classe "Setosa" est bien discriminée, sur les 50 individus auquel le modèle a assigné cette classe (première colonne du tableau), tous la portent effectivement. En revanche il est clair que les deux classes "Versicolor" et "Virginica" posent problème : parmi les 54 individus désignés par le classifieur comme "Versicolor", il y a 5 mauvais classements qui correspondent à la classe "Virginica". Ce résultat est bien connu en apprentissage, il n'est pas étonnant que nous le retrouvions avec les graphes tant le concept est facile à apprendre.

Pour avoir une meilleure idée de la structure de l'erreur dans la population mère, il est clair qu'il est plus approprié de mener l'analyse sur un échantillon test n'ayant pas servi à l'expansion du graphe. En effet, ce dernier lors de la construction a été optimisé pour augmenter la pureté de ses feuilles, lorsque l'on sur-apprend, seule la diagonale sera remplie sur le fichier d'apprentissage, laissant croire à une discrimination parfaite.

2.4.2 Estimation de l'erreur

Le taux d'erreur théorique, la probabilité que l'on se trompe si l'on applique le classifieur sur la population mère Ω, est un indicateur complètement objectif, comparable d'un algorithme à un autre. Il permet de quantifier le risque encouru en adoptant les décisions du classifieur. De plus, il est souvent d'usage d'utiliser cet indicateur pour choisir le modèle le plus approprié dans un domaine donné [Kohavi et John, 1997] [Scheffer et Herbrich, 1997]. Enfin, on peut également le mettre en oeuvre pour contrôler la combinaison de classifieurs d'origines différentes [Wolpert, 1992].

Pour une variable à prédire $Y(.)$, la formule du taux d'erreur est fort simple, il s'agit tout simplement d'une proportion. Soit $q(M,\omega)$ une fonction indicatrice telle que

$$q(M,\omega) = \begin{cases} 0 \; si \; M(\omega) = Y(\omega) \\ 1 \; si \; M(\omega) \neq Y(\omega) \end{cases}$$

L'erreur théorique s'écrit alors

$$\varepsilon = \frac{\omega \sum q(M,\omega)}{card(\Omega)}$$

Bien entendu, le calcul de la quantité ε est rarement possible, même si tous les individus de la population étaient accessibles, les coûts qui en résulteraient seraient exorbitants. Il nous faut donc trouver un estimateur qui présente des qualités de fiabilité suffisantes. Celles-ci peuvent être quantifiées grâce au biais et à la variance de l'estimateur statistique[8].

Pour une estimation $\widehat{\varepsilon}$, le biais statistique s'écrit

$$biais(\widehat{\varepsilon}) = \varepsilon - E[\widehat{\varepsilon}]$$

et la variance

$$Var(\widehat{\varepsilon}) = E[\widehat{\varepsilon} - E[\widehat{\varepsilon}]]^2$$

où l'opérateur $E[.]$ représente l'espérance mathématique. On dira qu'un indicateur est bon s'il a un biais et une variance faibles, l'intervalle de confiance qui en résulte permettra de quantifier avec rigueur la plage des valeurs possibles de ε.

L'estimation par resubstitution

Le premier estimateur toujours disponible est le taux d'erreur en resubstitution, il est calculé directement sur l'échantillon Ω^a $[n = card(\Omega^a)]$ ayant servi à l'apprentissage

$$\varepsilon_{resub} = \frac{\sum_{\omega \in \Omega^a} q(M,\omega)}{card(\Omega^a)}$$

8. biais et variance sont relatifs à l'estimation ici et n'ont qu'un rapport lointain avec la décomposition biais variance de l'erreur dûe à [Breiman, 1994]

Cet estimateur est largement décrié, il présente un biais d'optimisme ($biais(\varepsilon_{resub}) > 0$) trop souvent important. En effet, un des objectifs de l'apprentissage étant de constituer des sous-groupes les plus purs, il y a de fortes chances que le classifieur soit meilleur sur cet échantillon en particulier. A la limite, en ne constituant que des groupes composés de singletons, on est sûr que $\varepsilon_{resub} = 0$.

Le schéma classique apprentissage-validation

Afin de palier ce biais d'optimisme, très difficile à estimer de manière théorique car dépendant de la distribution des données, on propose généralement d'effectuer un traitement en deux étapes. On subdivise aléatoirement les observations disponibles en deux fractions :

- la première (Ω^a) sert classiquement à construire le classifieur,
- la seconde, l'échantillon de validation (Ω^v), est utilisé pour évaluer les performances du classifieur.

La formule adéquate est donc

$$\varepsilon_{valid} = \frac{\sum_{\omega \in \Omega^v} q(M, \omega)}{card(\Omega^v)}$$

Le taux d'erreur mesuré ainsi est théoriquement sans biais, en revanche sa variance

$$Var(\varepsilon_{valid}) = \frac{\varepsilon_{valid}(1 - \varepsilon_{valid})}{card(\Omega^v)}$$

est trop forte, il faut un grand nombre d'individus que [Catlett, 1991a] quantifie arbitrairement à 1000 pour la réduire à un niveau raisonnable.

Le succès de cette méthode repose sur la partition adéquate de l'échantillon total en fichiers d'apprentissage et de validation. Si les observations sont en nombre suffisant, il n'y a pas de problème la méthode peut construire un classifieur suffisamment riche pour approximer au mieux le concept. En réalité, ce schéma est rarement vérifié, le partage courant du fichier en 2/3-1/3 [Kohavi, 1995b] ou 70%-30% [Breiman *et al.*, 1984] est éminemment empirique, on constate généralement que le prédicteur est sous-dimensionné à cause de la réduction de l'effectif de l'échantillon d'apprentissage, l'estimateur ci-dessus est en réalité trop pessimiste.

Un dernier reproche que l'on pourrait adresser à cette méthode est qu'elle ne capte qu'une seule source de variabilité, celle en provenance de l'échantillon Ω^v. Rien ne permet ici d'appréhender la variabilité induite par l'échantillon d'apprentissage qui agit directement sur les performances du classifieur.

A cette fin, on a souvent rencontré dans la littérature de l'intelligence artificielle, le schéma d'apprentissage-validation répété. Le principe consiste à répéter v fois le processus suivant : tirer

au hasard n_1 individus parmi les n, de construire un classifieur M_1 que l'on validera sur le reste de l'échantillon $\Omega^{v,1}$ composé de $n - n_1$ individus. L'estimateur devient

$$\varepsilon_{valid\_répeté} = \frac{\sum_{\omega \in \Omega^{v,i}} q(M_i, \omega)/card(\Omega^{v,i})}{v}$$

L'estimateur ici est toujours sans biais, hélas sa variance est largement sous-évaluée parce qu'elle est calculée sur une série d'échantillons non indépendants, en effet ils ont plusieurs individus en commun. De plus, les distributions de probabilités asymptotiques de l'erreur empirique deviennent caduques dans ce contexte.

La validation croisée

La validation croisée propose une solution assez judicieuse au problème du recouvrement entre les fichiers de validation et de la trop faible taille de l'échantillon si l'on n'en utilise qu'un seul exemplaire. En vertu de la propriété selon laquelle plusieurs estimations sur des portions indépendantes d'un échantillon sont de variance plus faible qu'une estimation sur la totalité, cette technique procède à une répétition du couple "apprentissage-validation" mais en veillant à ce qu'il n'y ait aucun recouvrement entre les échantillons de validation [Stone, 1974].

L'algorithme est le suivant :

Subdiviser l'échantillon initial en s parties égales
Pour chaque portion u
  Former l'échantillon $\Omega^{a,-u}$ provenant des $(s-1)$ portions autres que u
  Créer un classifieur M_u à partir de $\Omega^{a,-u}$
  Calculer ε_u sur l'échantillon $\Omega^{a,u}$ correspondant à la $u^{ième}$ portion
FinPour
$\varepsilon_{cv} = \sum_u \frac{card(\Omega^{a,u})}{card(\Omega^a)} \varepsilon_u$
$Var(\varepsilon_{cv}) \approx \frac{\varepsilon_{cv}(1-\varepsilon_{cv})}{s}$

En procédant à une étude empirique très poussée du comportement de la validation croisée, [Kohavi, 1995b] est arrivé à la conclusion que la meilleure stratégie était une validation croisée avec $s = 10$. C'est le meilleur compromis avec un biais qui diminue avec l'augmentation du nombre de portions et la variance qui s'accroît dans le même temps. Afin de réduire cette dernière, il propose des améliorations similaires à celles de [Dietterich, 1996] à savoir la répétition de la validation croisée ou encore la stratification i.e respecter les distributions de classes inconditionnelles dans les différentes portions. Dans la pratique, [Kohavi, 1995a] pense que la solution la plus appropriée est une validation croisée stratifiée à 10 portions qui donne une bonne estimation de l'erreur et de son intervalle de confiance.

Il existe un cas particulier de la validation croisée que l'on rencontre très souvent dans la littérature : le "leave one out" que l'on connaît encore sous le nom de "jacknife". Il s'agit tout

simplement de fixer $s = n$, chaque portion n'est composé que d'un seul individu. Très long à mettre en oeuvre, surtout lorsque le fichier de données est de taille importante, il est pénalisé par une variance trop forte.

Le boostrap

Le boostrap est une technique empirique qui permet d'estimer l'espérance mathématique du biais d'optimisme de l'estimation par resubstitution [Efron, 1983] [Efron, 1986], elle n'est donc pas vraiment spécifique à un classifieur M, mais plutôt rattachée à une correction moyenne de l'erreur en resubstitution calculée sur les classifieurs produits par l'algorithme d'apprentissage.

Le principe est fondé sur B répétitions de l'apprentissage sur un échantillon $\Omega^{a,b}$ de taille n $[card(\Omega^a)]$ constitué par tirage aléatoire simple avec remise dans Ω^a. Cette technique est très gourmande en temps de calcul, en effet il est souvent recommandé de procéder à au moins une centaine de répétitions pour espérer avoir une bonne fiabilité. La formule de base s'écrit

$$\varepsilon_{boot} = \varepsilon_{resub} + \widehat{biais}(\varepsilon_{resub}) \tag{2.1}$$

Pour estimer le biais, nous utilisons à chaque réplication l'ensemble $\Omega^{a,-b}$ des observations de l'échantillon Ω^a qui n'ont pas été inclus dans $\Omega^{a,b}$, elle constitue en quelque sorte un échantillon test puisque le classifieur M_b ne pas été élaboré sur ces individus. Soient $\varepsilon(M_b, \Omega^{a,-b})$ le taux d'erreur calculé sur l'ensemble $\Omega^{a,-b}$ pour le classifieur M_b, et $\varepsilon(M_b, \Omega^a)$ le taux calculé sur l'ensemble du fichier d'apprentissage, alors [Efron, 1983] propose comme estimation du biais

$$\widehat{biais}(\varepsilon_{resub}) = \frac{\sum_b \varepsilon(M_b, \Omega^{a,-b}) - \varepsilon(M_b, \Omega^a)}{B}$$

Le pseudo-code associé est le suivant :

Pour B réplications
  Former l'échantillon $\Omega^{a,b}$ par un tirage de n éléments avec répetitions dans Ω^a
  Former l'échantillon $\Omega^{a,-b} = \Omega^a - \Omega^{a,b}$
  Créer un classifieur M_b à partir de $\Omega^{a,b}$
  Calculer les taux d'erreurs $\varepsilon(M_b, \Omega^{a,-b})$, $\varepsilon(M_b, \Omega^a)$
FinPour
$\widehat{biais}(\varepsilon_{resub}) = \frac{\sum_b \varepsilon(M_b, \Omega^{a,-b}) - \varepsilon(M_b, \Omega^a)}{B}$
Créer un classifieur M à partir de Ω^a
Calculer le taux d'erreur $\varepsilon_{resub} = \varepsilon(M, \Omega^a)$
$\varepsilon_{boot} = \varepsilon_{resub} + \widehat{biais}(\varepsilon_{resub})$

Il existe une autre approche de l'équation 2.1 qui semble donner plus satisfaction à l'auteur. Partant du constat que dans une réplication, sous l'hypothèse de tirages indépendants avec

remise, un individu a une probabilité

$$\begin{aligned}(1-\frac{1}{n})^n &\approx e^{-1}\\ &\approx 0.368\end{aligned}$$

de faire partie de $\Omega^{a,-b}$. Donc, il y a naturellement 0.632 chances pour qu'il appartienne à l'échantillon $\Omega^{a,b}$, [Efron, 1983] propose la correction suivante

$$\widehat{biais}_{0.632}(\varepsilon_{resub}) = 0.632 \times \frac{\sum_b [\varepsilon(M_b, \Omega^{a,-b}) - \varepsilon_{resub}]}{B}$$

ce qui permet de dériver une formule directe d'estimation de l'erreur

$$\varepsilon_{boot,0.632} = 0.632 \times \frac{\sum_b \varepsilon(M_b, \Omega^{a,-b})}{B} + 0.368 \times \varepsilon_{resub}$$

Il y a de nombreuses variantes autour de ces formulations originelles, la plupart pour corriger la propension à sous-estimer l'erreur en resubstitution initiale. En effet, pour certaines méthodes, $\varepsilon_{resub} = 0$, c'est le cas quand l'arbre est surdimensionné, ou encore pour des méthodes comme le plus proche voisin [Jain *et al.*, 1987].

Dans le cas où il s'applique, on constate généralement que le bootstrap a une faible variance par rapport à la validation croisée, il a un biais plus fort en revanche.

Paramètrage des modèles

Les algorithmes d'induction comportent pour la plupart un ou plusieurs paramètres à régler pour qu'ils s'adaptent au mieux au domaine d'étude, cela se traduit souvent par une préférence plus ou moins forte à la simplicité. Une attitude couramment rencontrée est de lancer l'algorithme, d'analyser l'erreur sur l'échantillon test, de régler les paramètres idoines, puis de relancer l'algorithme que l'on validera de nouveau. Ainsi de suite, jusqu'à ce que l'on soit satisfait du résultat. Ce va-et-vient peut être manuel, basé sur les intuitions de l'expert, ou automatique [John, 1994].

Trop souvent hélas, on utilise le même fichier pour régler les paramètres et mesurer l'erreur, sous prétexte que cet ensemble n'a pas servi à l'apprentissage. [Scheffer et Herbrich, 1997] montrent que le biais d'optimisme mesuré sur le fichier de réglage[9] augmente avec le nombre de paramètres et de répétitions, ces résultats peuvent être étendus sur les procédés de validation croisée.

Si l'on veut véritablement obtenir un estimateur réaliste de l'erreur, on devrait en fait adopter un schéma de double cross-validation dans le cadre du paramétrage automatique. Cela consiste à subdiviser le fichier en s portions toujours, puis sur le fichier d'apprentissage $\Omega^{a,u}$ procéder à

9. tuning set

une cross-validation pour la recherche des paramètres optimaux de l'algorithme sur un domaine d'étude.

Bien entendu, il est évident que les modifications manuelles ne sont pas réalisables dans ce cadre, on se contentera alors de subdiviser en trois l'échantillon initial [Salzberg, 1997] : apprentissage, réglage et validation. On utilise les deux premiers pour définir une série de paramètres, on jugera de leur pertinence après coup en confrontant tous les classifieurs produits sur l'échantillon de validation.

2.5 Comparaison de classifieurs

La section précédente était destinée à estimer au mieux les performances d'un classifieur, dans celle-ci nous nous attacherons à répondre à la question "le classifieur issu de l'algorithme A est-il meilleur que celui issu de B sur ces données?"(1). Nous discuterons plus loin de la généralisation à la question "A est-il meilleur que B?"(2)

Ce type de question peut être étendu à une modification dans un algorithme connu. Une méthode dans sa globalité peut être vue comme un puzzle où plusieurs pièces jouent des rôles essentiels, l'évaluation peut alors prendre la forme d'une analyse de variance à un facteur où l'on teste le rôle d'un élément précis. Nous pensons que dans la pratique c'est la seule méthode vraiment justifiée. En effet, pour des classifieurs différents nous ne contrôlons pas les éléments qui influent sur les performances de l'algorithme, les résultats ne sont qu'indicatifs et ne permettent pas d'analyser son comportement.

Dans un excellent papier qui survole la comparaison statistique de prédicteurs, [Dietterich, 1996] recense les principales sources d'aléas que l'on doit contrôler dans ces tests de comparaisons :

- le fichier de test, sachant qu'il est constitué aléatoirement, il y a toujours des chances pour qu'il soit atypique;
- remarque également valable pour ce qui est du fichier d'apprentissage, d'autant plus crucial que l'algorithme est sensible à ces variations, ce qui est le cas des graphes d'induction;
- la variabilité du classifieur lui-même. [Heath *et al.*, 1993a] par exemple utilisent un algorithme de recuit simulé pour trouver les meilleures partitions obliques dans l'espace de représentation, il est clair que l'arbre final change d'une itération à une autre sur le même fichier, cela est dû à la variance du nombre aléatoire utilisé pour approuver ou non une modification non-pertinente;
- enfin, le bruit contenu dans les données bien sûr agit sur le classifieur final.

Les tests doivent donc tenir compte de ces sources de variations, il est largement admis qu'il est nécessaire de procéder à plusieurs couples apprentissage-validation pour cela. Néanmoins,

dans le cas où l'on ne peut faire qu'un seul apprentissage, nous disposons quand même d'outils relativement puissants.

2.5.1 Par apprentissage unique

Nous ne disposons ici que d'un seul couple apprentissage-validation et de deux classifieurs M_A et M_B. Il est évident que l'on ne peut pas comparer avec une seule observation la complexité exprimée en nombre de noeuds de deux classifieurs tout simplement parce que l'on connaît pas sa loi de distribution. Il en est autrement en ce qui concerne le taux d'erreur. Pour chaque individu ω, on peut penser que la variable $q(M,\omega)$ suit une loi de Bernouilli avec les paramètres $Binomial(1,\varepsilon)$. Puisque l'on observe un grand nombre d'individus (du moins supérieur à 30), le théorème central limite s'applique directement et l'on admet que la variable ε_{valid} tend vers une loi normale $Normal(\varepsilon, \frac{\varepsilon(1-\varepsilon)}{card(\Omega^v)})$.

Test sur des proportions

Pour répondre à la question (1), on spécifie le test

$$\begin{cases} H_0 : \varepsilon_A = \varepsilon_B \\ H_1 : \varepsilon_A > \varepsilon_B \end{cases}$$

en utilisant les statistiques $\varepsilon_{valid,A}$ et $\varepsilon_{valid,B}$. Dans les faits, on constate que ce test est peu puissant, on lui préfère la spécification de McNemar [Dietterich, 1996] [Craven et Schavlik, 1996].

Test de McNemar

Le test de MacNemar [Everitt, 1977] s'intéresse à la structure de l'erreur. On forme le tableau de contingence suivant

n_{00}	n_{01}
n_{10}	n_{11}

où

- n_{00} : nombre de fois où M_A et M_B se sont trompés simultanément,
- n_{01} : nombre de fois où M_A s'est trompé mais pas M_B,
- n_{10} : nombre de fois où M_A ne s'est pas trompé alors que M_B oui,
- n_{11} : nombre de fois où M_A et M_B ont classé simultanément correctement.

Sous l'hypothèse de performances égales entre M_A et M_B, les quantités n_{10} et n_{01} ont la même valeur espérée qui est égale à $\frac{n_{01}+n_{10}}{2}$. Il reste alors à former la statistique du χ^2 à 1 degré de liberté qui s'écrit

$$\chi^2_{McNemar} = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}$$

La région critique pour un niveau de confiance α est tout simplement définie par les valeurs de $\chi^2_{McNemar}$ supérieures au point de pourcentage $1 - \alpha$ pour une loi du χ^2 à 1 degré de liberté. Ce test est relativement fiable, on détermine aisément le meilleur algorithme en comparant n_{10} et n_{01}.

2.5.2 Par apprentissage répété

Les solutions précédentes ne sont que des pis aller, il est clair que si l'on veut des comparaisons fiables, on doit procéder à plusieurs répétitions. Malgré les réticences de [Dietterich, 1996], nous pensons que la validation croisée est un cadre simple et viable pour les tests.

Test sur l'erreur

L'idée sous-jacente est d'exploiter les sous-groupes pour constituer une statistique formée de s observations (et non plus d'une seule comme ce serait le cas pour un seul échantillon test) car on peut spécifier chaque portion test $\Omega^{a,u}$ de manière à ce qu'elle contienne les mêmes individus pour les deux classifieurs. On effectue alors une comparaison deux à deux pour former un test de Student sur échantillons appariés dont voici la statistique [10]

$$d_{AB} = \frac{1}{s}\sum_{u=1}^{s}(\varepsilon_{u,A} - \varepsilon_{u,B})$$

Pour définir la région critique de rejet de l'hypothèse nulle, on utilise la quantité

$$t_{AB} = \frac{d_{AB}\sqrt{s}}{\sqrt{\frac{\sum_{u=1}^{s}(\varepsilon_{u,A}-\varepsilon_{u,B})^2}{s-1}}}$$

On rejette l'hypothèse nulle si

$$t_{AB} > t_{1-\alpha}(s-1)$$

où $t_{1-\alpha}(s-1)$ est la valeur lue dans la table de Student au point de pourcentage $1 - \alpha$.

10. on fait l'hypothèse ici que $card(\Omega^{a,u})$ est constant

Test sur la complexité

Le test de Student sur échantillons appariés est licite sur le taux d'erreur parce que nous avons vu plus haut qu'il était possible d'en formuler une distribution de probabilités approximativement normale. Ceci est plus problématique en ce qui concerne le nombre de noeuds dans le graphe. On sait que cette mesure est par nature discrète, il paraît extrêmement difficile de produire une approximation quelconque, ce qui condamne d'office toute approche paramétrique.

Fort heureusement, il existe dans la panoplie non-paramétrique un test qui se rapproche de la philosophie ci-dessus, à savoir l'utilisation de l'appariement des échantillons : le test de Wilcoxon [Aivazian *et al.*, 1986]. Il est hélas très peu utilisé dans la littérature de l'intelligence artificielle, pourtant il offre des qualités de robustesse et de souplesse qui dépassent largement le test de Student. En effet, il peut être appliqué sur n'importe quel type d'indicateur (taux d'erreur, nombre de noeuds, temps de calculs...) pourvu que celui-ci respecte une structure d'ordre.

Notre test d'hypothèses s'écrit :

$$\begin{cases} H_0 : \eta_A = \eta_B \\ H_1 : \eta_A > \eta_B \end{cases}$$

Nous restons toujours dans le cadre de la validation croisée en s portions. Soit $\eta_i(\Omega^{a,-u})$ le nombre de noeuds produits par l'algorithme i sur l'échantillon formé par les $(s-1)$ autres portions que u. Nous définissons la quantité suivante

$$\delta_u = \eta_A(\Omega^{a,-u}) - \eta_B(\Omega^{a,-u})$$

que l'on rangera dans l'ordre des valeurs croissantes de $|\delta_u|$: à chaque terme δ_u est affecté son rang $rg(\delta_u)$. En cas d'ex-aequo, nous affecterons un rang moyen.

On détermine deux quantités, la somme des rangs de différences positives et négatives

$$\begin{aligned} \Delta_+ &= \sum_{u:\delta_u>0} rg(\delta_u) \\ \Delta_- &= \sum_{u:\delta_u\leq 0} rg(\delta_u) \end{aligned}$$

Sous l'hypothèse nulle, lorsque le nombre d'observations est suffisamment important ($s > 25$), la quantité

$$\phi = \frac{\Delta_+ - \frac{s(s+1)}{4}}{\sqrt{\frac{s(s+1)(2s+1)}{24}}}$$

suit asymptotiquement une loi normale centrée et réduite.

Dans le cas de la validation croisée à 10 portions, ceci n'est pas applicable. Nous devons utiliser les valeurs exactes $\phi_{exact}(s)$ tabulées par Wilcoxon. La région critique est définie par

$$\min(\Delta_+, \Delta_-) < \phi_{exact}(s)$$

Pour $s = 10$, au niveau de confiance de 5% (resp. 1%), $\phi_{exact} = 8$ (resp. 3).

Nous remarquons que cette procédure ne cherche en aucune manière à calculer les caractéristiques numériques de l'indicateur à comparer. Comme le principal reproche adressé au schéma répété d'apprentissage-répétition par tirage aléatoire des individus (§ 2.4.2.0) était justement de sous-estimer la variance, on peut conclure que le test de Wicoxon lève cette limitation.

2.5.3 Un classifieur est-il meilleur qu'un autre en général?

Implicitement, la plupart des papiers où l'on propose de nouveaux algorithmes cherchent à montrer, au moins empiriquement, leur supériorité. On voit ainsi fleurir les grands tableaux de tests sur une trentaine, si ce n'est plus, de bases de données synthétiques ou réelles, où l'on consigne les moyennes de taux d'erreur et de nombre de règles en cross-validation. Rares sont ceux qui utilisent des tests statistiques pour évaluer la différence, certains [de Merckt et Quinlan, 1996] [Quinlan, 1996] proposent une série de ratios qui n'ont aucune signification, et pire encore font la moyenne de ces ratios pour montrer qu'une méthode est en général supérieure à une autre.

Nous ne pouvons que réprouver ce genre d'attitude. La seule démarche vraiment défendable serait de faire une comparaison des méthodes sur chaque fichier en utilisant les formulations précédentes, de créer une variable aléatoire γ_j qui prend la valeur 1 si l'algorithme A est meilleur que B sur la base j. Sous l'hypothèse nulle de performances égales, cette variable suit une loi de Bernouilli de paramètre 0.5. Le test, que l'on appelle test des signes, consiste alors à vérifier pour J bases que

$$\Gamma = \frac{1}{J} \sum_j \gamma_j$$

est bien significativement supérieur à 0.5.

C'est un test très simple, mais peut-on vraiment lui faire confiance? La condition sine qua non de son application est tout d'abord la représentativité de l'échantillon de bases de données utilisées, sont-elles vraiment issues d'un tirage aléatoire simple dans toutes les bases de données existantes. Il est clair que non, nous en avons déjà discuté plus haut. Plus judicieux peut-être serait de fixer un domaine d'étude (médecine, aéronautique..), sur lequel on pourra mieux délimiter la portée des fichiers utilisés.

La seconde remarque touche à la définition de la variable γ_j elle-même, elle dépend de la significativité du test utilisé pour comparer entre eux les algorithmes sur un domaine, selon que l'on soit plus ou moins permissif, la quantité Γ sera plus ou moins grande. Il est évident qu'il y a un phénomène d'incertitude supplémentaire dans la définition de la variable aléatoire que nous maîtrisons mal ici.

Pour ces raisons, nous nous abstiendrons tout le long de cette thèse de tirer des conclusions définitives à partir de nos expérimentations. Nous essaierons surtout de caractériser les fichiers pour lesquels telle ou telle méthode semble mieux se comporter.

2.6 Conclusion

Ce chapitre était surtout destiné à montrer combien il est délicat de comparer des méthodes sur la base d'évaluations empiriques. De nombreuses précautions sont nécessaires pour donner une signification statistique aux résultats. Les comparaisons n'ont vraiment de sens que pour un domaine d'étude donné, la généralisation à l'ensemble des domaines est très hasardeuse, pour ne pas dire illusoire.

Enfin, une grande prudence doit entourer toute étude fondée sur des données réelles en provenance de serveurs. Les résultats n'ont vraiment de sens que si l'on procède au préalable à une caractérisation minutieuse du fichier.

Deuxième partie

Les éléments de base de l'algorithme d'induction de graphes

Chapitre 3

Mesures de qualité des partitions

3.1 Introduction

Le choix du critère de sélection des attributs lors du partitionnement sur un noeud est certainement le domaine dans lequel les chercheurs ont été les plus prolixes ces dernières années [Safavian et Landgrebe, 1991]. En effet, l'enjeu est de taille : il s'agit de trouver, en vertu du principe du rasoir d'Occam [Blumer *et al.*, 1987] le modèle le plus simple qui soit cohérent sur les données. Ce problème est NP-complet [Hyafil et Rivest, 1976]. La stratégie la plus couramment utilisée pour construire les graphes de décision consiste alors à rechercher localement, sur chaque noeud, l'attribut qui induit le meilleur éclatement. L'idée sous-jacente étant qu'une succession de "bons" découpages nous rapproche au mieux du partitionnement optimal de taille minimum.

Prenons un exemple simple pour introduire nos propos. Soit un fichier de données fictif (Table 3.1) contenant 10 individus décrits à l'aide d'une variable à prédire Y prenant deux modalités $\{x, o\}$, et de deux variables prédictives booléennes (X_1, X_2), $\Omega^a = \{1, 2, \ldots, 10\}$.

Il y a deux manières de construire une partition cohérente, ne contenant que des noeuds purs, sur ces données. La première, représentée dans la figure 3.2 a utilisé successivement les variables X_1 et X_2 pour aboutir à *trois* sous-groupes purs. En revanche, si nous utilisons d'emblée la variable X_2, nous obtenons deux sous-groupes purs [figure 3.1]. Dans ce dernier cas, nous obtenons un graphe plus simple, constitués de noeuds terminaux avec des effectifs plus élevés en moyenne, donc de meilleure qualité lors de la prédiction. En utilisant une mesure qui nous permet de sélectionner la variable X_2 plutôt que X_1, lors du premier éclatement, nous aboutissons à une partition consistante minimale.

De très nombreuses études empiriques portant sur l'influence des mesures de qualité de partition lors de la construction d'un graphe d'induction ont été publiées ces dernières années. Dans un article fort célèbre, [Mingers, 1989b] a soutenu qu'un choix aléatoire des attributs sur un sommet permettait de construire des arbres de décisions présentant des taux de clas-

N°	Y	$\mathbf{X_1}$	$\mathbf{X_2}$
1	x	faux	vrai
2	x	faux	vrai
3	x	faux	vrai
4	x	vrai	vrai
5	x	faux	vrai
6	o	vrai	faux
7	o	vrai	faux
8	o	vrai	faux
9	o	vrai	faux
10	o	vrai	faux

TAB. 3.1 – *Fichier de 10 individus : une variable à prédire, deux attributs predictifs*

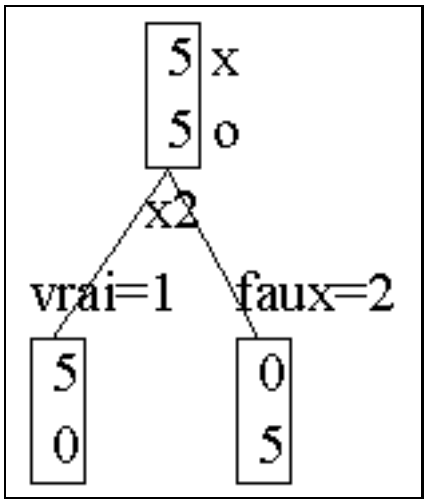

FIG. 3.1 – *Une partition pure par un arbre à un niveau*

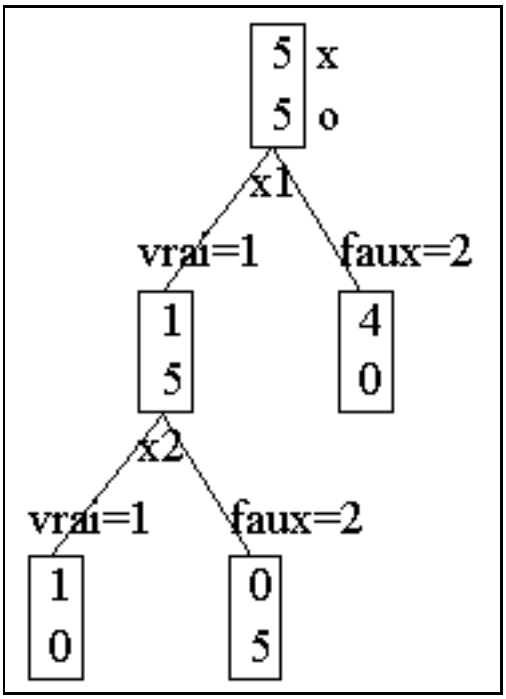

FIG. 3.2 – *Une partition pure par un arbre à deux niveaux*

sement identiques aux autres, mais de complexité plus élevée. [Buntine et Niblett, 1992] ont démonté point par point son argumentation. Ils ont montré que le protocole d'expérimentation était mal élaboré, les taux de succès en validation étaient en fait mesurés sur les fichiers ayant servi à l'élagage; de plus, les exemples pris étaient pour la plupart des fichiers exempts d'attributs non-pertinents. Néanmoins, de ce papier naquit un paradigme fort qui ne fut jamais démenti jusqu'à maintenant dans les études empiriques : pour peu qu'elle possède de *bonnes propriétés*, l'influence de la mesure de qualité de partition est manifeste sur la complexité du graphe d'induction construit, elle est en revanche faible sur ses performances en classification [Miyakawa, 1989][Lerman et Costa, 1996][Utgoff et Clouse, 1996][Mantaras, 1991].

Un des objectifs de ce chapitre est d'essayer de traduire le terme "bonnes propriétés". Nous utiliserons la formulation proposée par [Zighed, 1985], et nous montrerons pourquoi le taux d'erreur n'est pas un indicateur pertinent pour la construction des graphes d'induction. Puis, nous présenterons les différentes familles de mesures en reprenant partiellement les catégorisations proposées par [Ben-Bassat, 1987] et [Lerman et Costa, 1996] :

– *les mesures fondées sur la théorie de l'information* : la plus usitée dans la construction des arbres de décision est certainement l'entropie de Shannon [Shannon et Weaver, 1949]. Elle est notamment à la base de la fameuse méthode ID3 [Quinlan, 1979] qui contribua grandement à populariser les arbres en induction. Cette mesure est un cas particulier des entropies généralisées présentées par [Daroczy, 1970], son calcul pratique dépend es-

sentiellement de l'estimation des différentes probabilités d'apparition des classes utilisées [Zighed *et al.*, 1992].

- *les mesures fondées sur la distance entre les distributions de probabilités de classes* : le principe d'évaluation repose sur la séparabilité des classes, ce type de mesure était déjà présent dans des travaux fondateurs assez anciens [Bouroche et Tenenhaus, 1970][Friedman, 1977].

- *les mesures de dépendances* : elles sont souvent issues des indicateurs construits dans les deux catégories précédentes. Nous y aborderons les critères d'écarts à l'indépendance entre attributs catégoriels, et des mesures d'associations fondées sur le concept de la réduction de l'erreur [Olszak et Ritschard, 1995]. Ce dernier type d'approche propose une formulation non-symétrique qui permet de caractériser véritablement un processus de causalité.

- *les mesures fondées sur la comparaison par paires* [Marcotorchino, 1984] : on peut également retrouver ici les formulations précédentes. Nous nous cantonnerons à la famille de mesures proposées par [Lerman, 1992a][Lerman, 1992b].

Une constante que nous retrouverons tout le long de ce chapitre est l'existence d'une multitude d'interprétations autour des mêmes formules. Nous observerons notamment que l'interprétation en terme de corrélation est extrêmement fréquente, montrant ainsi parfaitement que la construction d'un classifieur peut être ramenée à la recherche de la combinaison des attributs prédictifs la plus corrélée avec la variable à prédire [Clark, 1990].

Nous ne procéderons point à des évaluations empiriques tant elles ont été déjà nombreuses, nous nous contenterons de présenter les axes d'études utilisés dans les plus importantes d'entre elles, et les résultats qui en ont découlé.

Mais auparavant, nous présentons dans la section qui suit les notations et les représentations de données sur lesquelles nous travaillons.

3.2 Notations et représentation des données

Soit Ω^a l'échantillon d'apprentissage, les valeurs $\{y_1, \ldots, y_K\}$ prises par la variable à prédire $Y(.)$ engendrent une partition sur cet ensemble tel que $n_{k.} = card(\{\omega \in \Omega^a \,/\, Y(\omega) = y_k\})$. Lorsque nous croisons $Y(.)$ avec l'attribut prédictif $X(.)$ prenant ses valeurs dans $\{x_1, \ldots, x_L\}$, nous définissons une autre partition de l'ensemble initial tel que $n_{kl} = card(\{\omega \in \Omega^a \,/\, Y(\omega) =$

y_k *et* $X(\omega) = x_l\}$). Nous pouvons ainsi définir le tableau de contingence suivant

	$X = x_1$	.....	$X = x_l$	.....	$X = x_L$	$\sum$
$Y = y_1$						
...			$\vdots$			
$Y = y_k$		...	n_{kl}	...		$n_{k.}$
...			$\vdots$			
$Y = y_K$						
$\sum$			$n_{.l}$			n

Dans ce qui suit, nous noterons indifféremment T ou $T_{Y/X}$ ce tableau, T_Y représentera sa marge droite. Nous définissons également les probabilités suivantes :

- $p_{k.} = \Pr(Y = y_k)$, il est estimé usuellement par $\frac{n_{k.}}{n}$
- $p_{.l} = \Pr(X = x_l)$, il est estimé usuellement par $\frac{n_{.l}}{n}$
- $p_{k/l} = P(Y = y_k / X = x_l)$, la probabilité conditionnelle estimée usuellement par $\frac{n_{kl}}{n_{.l}}$. Afin d'alléger l'écriture, et tant qu'il n'y aura pas d'ambiguïté, nous le noterons simplement p_{kl}

3.3 Propriétés "désirées" des mesures

3.3.1 Inefficacité du taux d'erreur

L'objectif de l'induction étant de construire un classifieur qui se trompe le moins en généralisation, on pouvait penser que le meilleur indicateur que l'on devrait optimiser serait le taux d'erreur mesuré en resubstitution [Lubinsky, 1993] [Lubinsky, 1994]. En effet, dans le cadre d'un tirage aléatoire simple et d'une matrice de coûts symétrique, l'affectation de la décision à un noeud revient à choisir la classe qui est la plus représentée, ce qui minimise le coût moyen de mauvais classement. Son expression est la suivante :

$$err(T) = \sum_{l=1}^{L} \frac{n_{.l}}{n} \times (1 - \frac{\max_k(n_{kl})}{n})$$

Dans la pratique, on se rend compte que cet indicateur n'est pas approprié du fait même qu'il ne prend en compte que la quantité de l'erreur, sans en restituer sa structure en terme probabiliste. [Breiman *et al.*, 1984] pensent que hors du cadre d'équidistribution des probabilités a priori d'apparition des classes, l'affectation obligatoire d'une conclusion à un sommet pour mesurer l'erreur est préjudiciable à la construction globale de l'arbre de décision.

Nous illustrons notre propos à l'aide d'un petit exemple repris chez [Utgoff et Clouse, 1996]

	x_1	x_2
y_1	9	1
y_2	100	900

Dans le cadre d'un tirage aléatoire simple, la distribution a priori des classes estimées sur les fréquences est bien $\widehat{p}_{1.}\widehat{p}_{2.} = \frac{10}{1010}\frac{1000}{1010} = 0.00990.9901$. Après la subdivision en deux sommets enfants, les distributions a posteriori deviennent respectivement $\widehat{p}_{11}\widehat{p}_{21} = \frac{9}{109}\frac{100}{109} = 0.08250.9175$ et $\widehat{p}_{12}\widehat{p}_{22} = \frac{1}{901}\frac{900}{901} = 0.00110.9989$. Connaissant donc la subdivision induite par une variable $X(.)$, nous améliorons notre connaissance de la distribution des classes (y_1, y_2), bien que cela n'affecte en rien le choix de la conclusion sur chaque sommet. En utilisant le taux d'erreur en resubstitution, nous observons que cette situation n'est pas du tout prise en compte. En choisissant la classe y_2 comme conclusion sur tous les sommets, nous obtenons respectivement 0.0099 et $\frac{109}{1010}0.0825 + \frac{901}{1010}0.0011 = 0.0099$ avant et après le partitionnement.

Le second argument qui milite en défaveur de $err(T)$ est sa forte tendance à l'optimisme lors de l'apprentissage. Sachant que l'on cherche le graphe qui minimise le taux d'erreur en classement, en utilisant le taux d'erreur en apprentissage nous nous exposons à la construction de sommets trop rapidement spécialisés, avec de petits effectifs.

Certes, les travaux de [Kalkanis, 1993], fondés sur une estimation pessimiste du taux d'erreur par la borne haute de son intervalle de confiance, corrige en partie cette propension. Il reste que l'obligation de choisir localement une conclusion, que nous suspectons d'autant plus néfaste que le nombre de classes à prédire augmente, hypothèque grandement l'efficacité de cette approche.

3.3.2 Propriétés désirées d'une "bonne" mesure

Ces premières remarques nous amènent à nous poser la question de savoir s'il existe une caractérisation d'une "bonne" mesure de qualité de partitions. [Zighed *et al.*, 1992] a proposé une série de propriétés que doivent présenter l'indicateur utilisé lors de la sélection des attributs sur un sommet du graphe d'induction. Ces propriétés sont inspirées de celles des entropies généralisées de [Daroczy, 1970] décrites dans [Wehenkel, 1996]. Notons que [Breiman *et al.*, 1984] dans la monographie CART proposent des propriétés similaires.

Ces caractéristiques reposent sur quatre principes que l'on pourrait décrire comme suit :

- la partition étant induite par des variables toutes nominales, la commutation des colonnes (des lignes) ne doit pas modifier la valeur de l'indicateur : c'est le principe de *symétrie*.
- même si nous n'utilisons pas le taux d'erreur, le principal objectif est toujours de construire des sous-groupes purs du point de vue de la variable à prédire, nous retrouverons cette exigence dans les propriétés de *maximalité* et de *minimalité*.

- la même partition construite sur un échantillon de plus grande taille, toutes choses égales par ailleurs, doit être qualifiée par une valeur plus élevée (ou plus faible selon que l'on minimise ou maximise la mesure) de l'indicateur. Ce principe est certainement un des plus importants introduits par [Zighed *et al.*, 1992] dans la construction des graphes d'induction. La plupart des indicateurs en effet sont fondés sur la manipulation de probabilités approximées simplement par la fréquence, l'estimateur du maximum de vraisemblance. Il est clair que sa variance, donc sa fiabilité, est inversement proportionnelle à la taille de l'échantillon qui sert à l'estimer. Hélas, les mesures utilisées ne traduisent pas cette réduction de variance lorsque la taille de l'échantillon augmente. Nous verrons dans les prochaines sections que l'utilisation d'une autre estimation des probabilités est une voie de recherche fructueuse. La propriété associée est la *sensibilité à la taille de l'effectif.*

- la mesure doit pouvoir comparer deux partitions de tailles inégales, induites par des attributs prédictifs prenant leurs valeurs dans des ensembles de cardinaux différents. C'est également un des sujets de recherche les plus ardents dans la construction des graphes. Les mesures utilisées dans les premiers algorithmes favorisaient exagérément les attributs multivalués (à plusieurs modalités) et conduisaient à la construction de graphes trop larges avec de nombreux sommets de petites tailles. Les débats et les travaux ont été nombreux en la matière, nous y consacrerons une section entière dans ce chapitre. Ce principe a été traduit par la propriété de *fusion*. Notons que cette notion est corollaire de la notion de sensibilité aux effectifs puisque la fusion de sommets de distributions identiques produit un sommet de plus grande taille mais de même distribution. Dans la construction d'arbres binaires où les modalités de l'attribut prédictif sont rassemblées en deux groupes disjoints [Fayyad et Irani, 1991], cette propriété n'a plus lieu d'être.

Nous donnons ici les expressions mathématiques des propriétés telles qu'elles ont été décrites par [Zighed *et al.*, 1992], nous les utiliserons par la suite pour vérifier si telle ou telle mesure les satisfait véritablement.

Soit $T = (T_1|\ldots|T_L)$ un tableau de contingence constitué à partir d'une partition en L groupes, $\Psi(T)$ une mesure de qualité de partition.

Proposition 1 ***Symétrie***

La mesure $\Psi(T)$ est invariante au regard des permutations des colonnes de T.

Proposition 2 ***Minimalité***

Si $\forall l \in \{1,\ldots,L\}$, $\exists k \in \{1,\ldots,K\}$ tel que $n_{kl} = n_{.l}$ alors $\Psi(T)$ est minimal.

Proposition 3 ***Maximalité***

Si $\forall l \in \{1, \ldots, L\}$, $n_{1l} = \cdots = n_{Kl}$ alors $\Psi(T)$ est maximal

Proposition 4 ***Sensibilité à la taille de l'effectif***

Si on multiplie tous les effectifs de T par une quantité m $(m > 1), T' = (m \times T_1 | \ldots | m \times T_L)$, alors $\Psi(T') < \Psi(T)$

Proposition 5 ***Fusion***

Soit $T = (T_1 | \ldots | T_i | \ldots | T_j | \ldots | T_L)$ un tableau de contingence. S'il existe un doublet i et j avec $T_i = s \times T_j$, alors la fusionnée $T'' = (T_1 | \ldots | T_i + T_j | \ldots | T_L)$ est telle que $\Psi(T'') < \Psi(T)$.

Il existe une dernière propriété dans [Zighed *et al.*, 1992] qui tient plus de la commodité calculatoire que de l'évaluation de la partition. L'auteur stipule que la variation de l'indicateur suite à une subdivision (ou une fusion) locale ne doit dépendre que des modifications sur les sommets concernés. L'objectif sous-jacent ici est de disposer d'une mesure qui permet d'apprécier la partition globale sans avoir à recalculer totalement l'indicateur après chaque éclatement. [Breiman *et al.*, 1984] a montré que pour les mesures issues de la théorie de l'information, exprimées sous la forme d'une moyenne pondérée d'entropie (ou d'indicateur de pureté si l'on se réfère à la terminologie des auteurs), un gain local est égal, à un coefficient multiplicatif près, au gain global. Nous ne l'utiliserons pas pour apprécier les différentes mesures car on peut facilement se passer de cette commodité en utilisant des astuces de programmation lors de l'implémentation des méthodes de construction de graphes.

Enfin, nous remarquons qu'à l'origine, les propriétés 4 et 5 étaient définies avec des inégalités au sens large. Nos derniers travaux nous ont permis d'une part de définir une mesure [Zighed et Rakotomalala, 1996a], d'autre part de trouver une série d'indicateurs, qui remplissent les inégalités au sens strict [Rakotomalala et Zighed, 1997].

3.4 Mesures fondées sur la théorie de l'information

3.4.1 Gain informationnel fondé sur l'entropie de Shannon

Développé par [Shannon et Weaver, 1949], la théorie de l'information formalise la notion d'information en terme d'entropie. Elle est à la base de très nombreux travaux en reconnaissance de formes [Watanabe, 1981], nous la retrouvons dans de très nombreuses méthodes de construction d'arbres de décision [Wang et Suen, 1984] [Casey et Nagy, 1984] [Hartmann *et al.*, 1982] [Sethi et Chatterjee, 1977], dont le très célèbre ID3 de [Quinlan, 1979].

Les éléments qui sont à la base de ce succès sont certainement une cohérence théorique forte et une interprétation aisée.

Entropie

L'entropie est une mesure d'incertitude relative aux valeurs prises par une variable aléatoire. Prenons le cas de Y prenant ses valeurs dans $\{y_1, ..., y_K\}$, la quantité de bits nécessaires pour connaître la classe d'une observation est égale à

$$S(T_Y) = -\sum_{k=1}^{K} p_{k.} \log_2(p_{k.})$$

De fait, dans tout le reste de notre travail, sauf mention contraire, tous les logarithmes seront à base 2. Dans la pratique, puisque

$$\lim_{a \to 0} a \log a = 0$$

nous fixerons $0 * \log(0) = 0$ dans l'implémentation de cette méthode [Magerman, 1994].

Il y a une description très précise de l'origine et de l'élaboration de cette mesure dans [Volle, 1985], nous constatons qu'elle répond parfaitement aux propriétés de *minimalité* et de *maximalité*.

Entropie conditionnelle

Nous voulons maintenant calculer la quantité d'information nécessaire pour connaître la valeur prise par Y sachant la modalité prise sur un attribut prédictif $X(.)$ prenant ses valeurs dans $\{x_1, ..., x_L\}$. Le nombre de bits nécessaires pour connaître $Y(.)$ sachant la valeur x_l de $X(.)$ est égal à

$$S(T_{Y/x_l}) = -\sum_{k=1}^{K} p_{kl} \log_2(p_{kl})$$

en moyenne donc, pour connaître la valeur de $Y(.)$ en connaissant la valeur prise par $X(.)$, nous obtenons

$$S(T_{Y/X}) = -\sum_{l=1}^{L} p_{.l} \sum_{k=1}^{K} p_{kl} \log_2(p_{kl})$$

Gain informationnel

Avec ces deux premières quantités, nous pouvons calculer la réduction de l'information nécessaire pour connaître la valeur prise par $Y(.)$ en passant de la distribution a priori à la distribution a posteriori dégagée à partir de l'attribut $X(.)$, c'est la notion de gain informationnel.

$$\Delta S(T_{Y/X}) = -\sum_{k=1}^{K} p_{k.} \log_2(p_{k.}) - (-\sum_{l=1}^{L} p_{.l} \sum_{k=1}^{K} p_{kl} \log_2(p_{kl})) \tag{3.1}$$

Dans la pratique, on utilise l'estimation des probabilités par la fréquence pour estimer cette quantité sur un échantillon $[\Delta \widehat{S}(T_{Y/X})]$. Nous discuterons plus loin des variations autour de ce thème.

3.4.2 Autre interprétation de la formule $\Delta S(T_{Y/X})$

L'interprétation de la formule 3.1 est à n'en pas douter celle qui est la mieux connue. Toutefois, avec le développement de la théorie bayesienne dans la construction des arbres de décision [Buntine, 1992], puis plus généralement dans la construction des graphes d'induction [Oliveira et Sangiovanni-Vincentelli, 1995]. On s'est rendu compte que cette spécification n'était en réalité qu'une fraction d'une formulation plus globale qui consiste à rechercher le classifieur le plus probable connaissant le fichier d'apprentissage. De fait, optimiser l'équation 3.1 revenait tout simplement à chercher la variable $X_i(.)$ qui maximise le rapport de vraisemblance [Buntine, 1991]

$$\max_{i=1,\ldots,p} \frac{P(T/H_0)}{P(T/H_1)} \tag{3.2}$$

L'hypothèse nulle H_0 correspond à la situation où la distribution des classes est restée identique à celle du sommet initial, quel que soit le sommet enfant sur lequel nous nous trouvons. Dans l'hypothèse alternative H_1, il y a une modification de ces distributions. Puisque nous voulons mettre en exergue des groupes dans lesquels certaines classes se distinguent, donc présentant un différentiel de distribution fort par rapport à la situation précédente, nous choisissons bien l'attribut qui correspond à l'équation 3.2.

Il existe une pléthore de démonstrations montrant l'équivalence entre les équations 3.2 et 3.1. Nous reconstruirons dans cette section celle de [Munteanu, 1996], qui curieusement aboutit à une autre formulation, proche du gain ratio [Quinlan, 1986b]. En reconstruisant la démonstration, nous n'avons pas retrouvé ses résultats mais plutôt ceux, bien connus maintenant, de [Buntine, 1992] et [Wehenkel, 1992].

Sans pour autant restreindre la portée de ses résultats, [Munteanu, 1996], qui attribue la paternité de ses travaux à [Jaynes, 1994], propose la solution pour un problème à deux classes(exemples - contre-exemples) et une partition binaire (droite - gauche). Nous illustrons cette situation (figure) avec les probabilités associées :

- $p_{1.}$ et $p_{2.}$ sont respectivement les probabilités théoriques d'obtenir les exemples et contre-exemples dans le sommet initial;
- sous l'hypothèse nulle (H_0) de non-pertinence de l'attribut prédictif $X(.)$, nous aurons simultanément la même distribution de classes $(p_{1.}, p_{2.})$ dans les deux sommets enfants;
- en revanche, si l'attribut induit une meilleure connaissance des probabilités prises par la variable à prédire (H_1), nous aurons deux distributions distinctes dans les sommets enfants : (p_{11}, p_{21}) à gauche et (p_{12}, p_{22}) à droite;
- $p_{.1}$ (resp. $p_{.2}$) est la probabilité qu'un individu aille dans la branche de gauche (resp. droite).

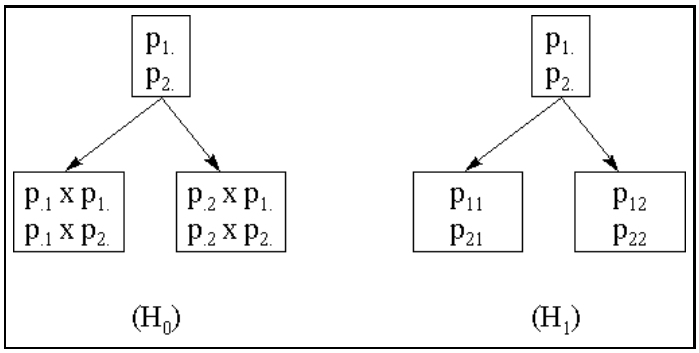

FIG. 3.3 – *Hypothèses pour tester la présence significative d'une structure dans les données*

L'échantillon d'apprentissage amène n individus, avec $n_{1.}$ exemples et $n_{2.}$ contre-exemples. La subdivision est représentée dans le tableau de contingence T suivant

	Gauche	Droite
Exemples	n_{11}	n_{12}
Contre-Ex.	n_{21}	n_{22}
$\sum$	$n_{.1}$	$n_{.2}$

Le critère de vraisemblance permet de décider quelle est l'hypothèse la plus crédible compte tenu de ces observations

$$O(H_1/H_0) = \frac{P(T/H_1)}{P(T/H_0)}$$

Comment calculer ces probabilités?

Selon l'auteur, le tableau T sous l'hypothèse alternative est le fruit de la conjonction de trois propositions élémentaires :

D parmi les n individus : $(n_{.1})$ observations sont allées sur la branche de gauche, $(n_{.2})$ sont allées sur la branche de droite;

D_g parmi les $(n_{.1})$ dans la branche de gauche, il y a n_{11} exemples et n_{21} contre-exemples;

D_d parmi les $(n_{.2})$ dans la branche de droite, il y a n_{12} exemples et n_{22} contre-exemples.

Dès lors la probabilité $P(T/H_1)$ devient

$$P(T/H_1) = P(D/H_1) \times P(D_g/H_1) \times P(D_d/H_1)$$

Les individus étant le fruit d'un tirage aléatoire dans la base d'apprentissage, nous pouvons calculer chacune de ces quantités sur la base d'un tirage binomial utilisant les probabilités définies ci-dessus :

$$P(T/H_1) \quad = \quad nn_{.1} \times p_{.1}^{n_{.1}} \times p_{.2}^{n_{.2}}$$

$$\times n_{.1} n_{11} \times p_{11}^{n_{11}} \times p_{21}^{n_{21}}$$
$$\times n_{.2} n_{22} \times p_{21}^{n_{21}} \times p_{22}^{n_{22}}$$

Sous l'hypothèse nulle, arguant que tous les individus proviennent d'un processus unique (la distribution des classes est la même dans les trois sommets), [Munteanu, 1996] considère le tableau T comme issu d'un tirage exhaustif dans une urne contenant n observations, les distributions sont donc maintenant hypergéométriques. Ce changement du mode de tirage est assez mystérieux. En effet, entre les deux hypothèses, seules les distributions sous-jacentes soumises au test ont été modifiées, ceci n'influe en rien sur le mode de tirage des individus. A notre sens, il est plus approprié de décrire la succession de tirages binomiaux en injectant les probabilités appropriées.

$$\begin{aligned} P(T/H_0) &= n n_{.1} \times p_{.1}^{n_{.1}} \times p_{.2}^{n_{.2}} \\ &\quad \times n_{.1} n_{11} \times p_{1.}^{n_{11}} \times p_{2.}^{n_{21}} \\ &\quad \times n_{.2} n_{22} \times p_{1.}^{n_{21}} \times p_{2.}^{n_{22}} \end{aligned}$$

En passant au logarithme du rapport de vraisemblance, nous obtenons :

$$\begin{aligned} L[O(H_1/H_0)] &= -[(n_{11}+n_{12})\log(p_{1.}) + (n_{21}+n_{22})\log(p_{2.})] \\ &\quad -[-[n_{11}\log(p_{11}) + n_{21}\log(p_{21})] - [n_{12}\log(p_{1.2}) + n_{22}\log(p_{22})]] \end{aligned} \tag{3.3}$$

Dans la pratique, nous estimons les probabilités à l'aide de la fréquence. Si nous divisons la quantité de l'équation 3.3 par n, l'expression devient

$$\begin{aligned} \frac{L[O(H_1/H_0)]}{n} &= -[\frac{n_{1.}}{n}\log(\frac{n_{1.}}{n}) + \frac{n_{2.}}{n}\log(\frac{n_{2.}}{n})] \\ &\quad -[-\frac{n_{.1}}{n}(\frac{n_{11}}{n_{.1}}\log(\frac{n_{11}}{n_{.1}}) + \frac{n_{21}}{n_{.1}}\log(\frac{n_{21}}{n_{.1}})) \\ &\quad -\frac{n_{.2.}}{n}(\frac{n_{12}}{n_{.2}}\log(\frac{n_{12}}{n_{.2}}) + \frac{n_{22}}{n_{.2}}\log(\frac{n_{22}}{n_{.2}}))] \end{aligned}$$

qui est le gain d'entropie calculé sur une partition binaire dans un problème à deux classes.

La généralisation à une variable à prédire comportant K classes et un attribut prédictif à L modalités est assez simple, au tirage binomial est substitué un tirage multinomial [Aivazian *et al.*, 1986]. De fait, nous pouvons affirmer que chercher un attribut $X(.)$ qui maximise le rapport de vraisemblance lors de la subdivision d'un sommet revient à chercher le même attribut $X(.)$ qui maximise le gain d'entropie de Shannon.

$$L[O(H_1/H_0)] = n \times \Delta\widehat{S}(T_{Y/X})$$

Ces calculs ne remettent pas en question les travaux de [Munteanu, 1996], ils montrent surtout que la diversité des mesures fondées sur les formulations bayésienne repose essentiellement du choix de la distribution a priori des modèles que l'on s'est fixé. Lorsqu'elle est judicieusement choisie, nous pouvons effectivement retomber sur une formulation proche du gain ratio[Quinlan, 1993a]. En revanche, faire porter cette propriété à la seule expréssion de la maximisation de la vraisemblance paraît assez hasardeux. En effet, dans le cadre d'une distribution uniforme, donc s'exprimant par une fonction de densité constante [Buntine, 1992], le meilleur modèle sera celui qui maximise le gain informationnel, à savoir la méthode ID3 de [Quinlan, 1979].

3.4.3 Calcul du gain informationnel sur un échantillon d'apprentissage : variations autour de l'estimation des probabilités

Estimation des probabilités à l'aide de la fréquence empirique

Dans la plupart des applications, notamment dans la construction des graphes d'induction, l'approximation des probabilités utilisées est tout simplement la fréquence empirique qui n'est autre que l'estimateur du maximum de vraisemblance :

$$\begin{aligned} \widehat{p}_{k.} &= \frac{n_{k.}}{n} \\ \widehat{p}_{kl} &= \frac{n_{kl}}{n_{.l}} \end{aligned}$$

On peut lire directement dans le tableau de contingence formé à partir des observations de l'échantillon d'apprentissage. Le gain empirique calculé sur l'échantillon est alors

$$\Delta\widehat{S}(T_{Y/X}) = -\sum_{k=1}^{K} \frac{n_{k.}}{n} \log_2(\frac{n_{k.}}{n}) - (-\sum_{l=1}^{L} \frac{n_{.l}}{n} \sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}} \log_2(\frac{n_{kl}}{n_{.l}}))$$

[Zighed, 1985] a beaucoup critiqué cette approche, en effet elle travaille uniquement sur des proportions, sans appréhender le nombre d'individus associés à chaque sommet. La propriété de sensibilité aux effectifs n'est pas du tout respectée ici. Cette restriction est levée lorsque nous utilisons l'estimation laplacienne des probabilités.

Estimation des probabilités à l'aide de la loi de succession de Laplace

Il existe de nombreuses interprétations de la fameuse loi de succession de Laplace [Wallace et Freeman, 1987] [Cestnik, 1990] [Jaynes, 1994]. La plus séduisante à notre sens, car la plus simple, est l'interprétation bayesienne que nous allons reproduire sur un exemple très simple tiré de [Jacquard, 1992].

Soit une urne contenant des boules rouges et des boules bleues. Nous voulons savoir à partir d'un tirage aléatoire simple avec remise la proportion de boules rouges qu'elle contient, ce qui

revient à estimer la probabilité p d'obtenir une boule rouge sur un tirage. En l'absence d'informations sur la composition de l'urne, on peut admettre le postulat bayésien selon lequel p peut prendre n'importe quelle valeur entre 0 et 1, avec une distribution uniforme. En effet, a priori, si nous ne connaissons rien du phénomène, il n'y a aucune raison de croire que p est proche de telle ou telle valeur. Nous adoptons donc la distribution a priori uniforme de fonction de densité

$$f(p) \begin{cases} = 0, pour\ (p < 0)\ ou\ (p > 1) \\ = 1, pour (0 \leq p \leq 1) \end{cases}$$

Après avoir effectué n tirages et obtenu r boules rouges, nous voulons inférer sur la nouvelle loi de p. Le théorème de Bayes discret nous enseigne l'estimation suivante

$$P(E_i/A) = \frac{P(E_i) \times P(A/E_i)}{\sum_{i=1}^{n} P(E_i) \times P(A/E_i)}$$

où $P(E_i)$ est la probabilité a priori d'avoir $i = r$, et $P(A/E_i)$ la probabilité d'obtenir r lors d'un tirage avec remise sachant que nous avions i boules rouges dans l'urne.

En nous positionnant dans le cadre continu, la distribution a posteriori de p s'écrit

$$g(p) = \frac{f(p)\, P(A/p)}{\int_0^1 f(p)\, P(A/p)\, dp}$$

$P(A/p)$ devient ici la probabilité de tirer r boules rouges sachant que la probabilité d'en tirer une est de p. Dans le cadre d'un tirage aléatoire simple avec remise, nous retombons sur une loi binomiale $B(n, p)$, dont la fonction de densité est bien connue [Ventsel, 1973]

$$P(r; n, p) = nrp^r(1-p)^{n-r}$$

Ainsi, $g(p)$ peut être aisément calculé

$$g(p) = \frac{p^r(1-p)^{n-r}}{\int_0^1 p^r(1-p)^{n-r}\, dp}$$

qui est tout simplement une loi bêta de paramètre $\beta(r+1, n-r+1)$, dont l'espérance mathématique s'écrit

$$\widetilde{p} = \frac{r+1}{n+2}$$

En passant à la généralisation sur une variable aléatoire pouvant prendre K valeurs, et une distribution a priori suivant une loi bêta symétrique de paramètre λ (la loi uniforme est une loi bêta symétrique de paramètre $\lambda = 1$), nous obtenons l'expression générale de la loi de succession de Laplace

$$\widetilde{p} = \frac{r+\lambda}{n+K\lambda}$$

C'est cette expression que [Zighed *et al.*, 1992] utilise dans les premiers algorithmes SIPINA d'induction de graphes. L'estimation du gain informationnel intègre ces nouvelles spécifications

$$\Delta\widetilde{S}(T_{Y/X}) = -\sum_{k=1}^{K}\frac{n_{k.}+\lambda}{n+K\lambda}\log_2(\frac{n_{k.}+\lambda}{n+K\lambda}) - (-\sum_{l=1}^{L}\frac{n_{.l}}{n}\sum_{k=1}^{K}\frac{n_{kl}+\lambda}{n_{.l}+K\lambda}\log_2(\frac{n_{kl}+\lambda}{n_{.l}+K\lambda}))$$

Le passage à cette estimation des probabilités confère de nouvelles propriétés aux mesures d'entropie. [Zighed *et al.*, 1992] montre que les mesures ainsi définies répondent aux critères de qualité décrits dans le paragraphe 3.3.2. [Rabaseda, 1996] a récemment repris la démonstration pour l'entropie quadratique.

3.4.4 Extensions aux formules généralisées d'entropie

L'entropie de Shannon n'est finalement qu'un cas particulier des fonctions d'entropie généralisées décrites dans [Aczel et Daroczy, 1975]. L'entropie de type β, où β est positif et différent de 1, est définie comme suit

$$D(T_Y) = \sum_{k=1}^{K} p_{k.}\, u^{\beta}(p_{k.})$$

avec $u\beta(p_{k.})$ une fonction d'incertitude strictement décroissante [Wehenkel, 1996] définie par

$$u^{\beta}(p_{k.}) = \frac{2^{\beta-1}}{2^{\beta-1}-1}(1-p_{k.}^{\beta-1})$$

On retrouve bien ici, en fonction des différentes déclinaisons de β, les différentes fonctions d'entropies :

- pour $\beta = 2$, nous obtenons l'entropie quadratique, connue également sous la dénomination d'indice de Gini, utilisée notamment dans les méthodes CART [Breiman *et al.*, 1984] et SIPINA (avec une estimation bayésiennes des probabilités) [Zighed *et al.*, 1992]. Cette forme d'entropie peut s'interpréter en terme de variance [Light et Margolin, 1971], [Heath *et al.*, 1993b] ont construit un indicateur analogue en calculant la variance des codes (e.g $y_1 = 1, y_2 = 2, \ldots$) affectés à chaque modalité de la variable à prédire;

- pour $\beta = 1$, $u^{\beta}(p_{k.})$ n'est pas défini. En revanche, au passage à la limite nous obtenons l'entropie de Shannon [Daroczy, 1970] puisque

$$\lim_{\beta\to 1} u^{\beta}(p_{k.}) = -\log_2(p_{k.})$$

Enfin, l'expression du gain d'entropie peut être largement étendue en passant par différentes formules de normalisation qui enrichissent et modifient leurs propriétés [Wehenkel, 1996].

3.5 Mesures fondées sur la distance

Cette famille de mesures rassemble les indicateurs qui quantifient la distance entre distributions de probabilités. Elle est réservée à la construction d'arbre binaire. Lors de l'utilisation d'attributs prédictifs comportant plus de deux modalités, nous devons effectuer des regroupements.

3.5.1 Les distances entre distribution de probabilité

Le passage à la discrimination binaire permet d'analyser le problème sous l'angle du calcul de la distance entre deux points plongés dans $\mathbb{R}^K$. En définissant les centres de gravité, nous pouvons calculer les différentes inerties, notamment l'inertie inter-groupes que l'on interprète alors comme la distance entre les deux distributions de probabilités de classes.

Construction du critère

A chaque individu $\omega \in \Omega^a$, nous associons un vecteur ligne Y de $\mathbb{R}^K$, qui est composé de zéros sauf à la $k^{ème}$ colonne lorsque la variable à prédire $Y(.)$ est égal à y_k, dans ce cas la valeur est 1. Les individus forment un nuage de points dans $\mathbb{R}^K$. En procédant à une subdivision binaire, nous isolons deux nuages de points.

Nous définissons les quantités suivantes :

1. le centre de gravité g du nuage initial de dimension $(1 \times K)$
$$g = \frac{1}{n} \sum_{\omega \in \Omega^a} Y(\omega)$$
2. le centre de gravité g_1 (resp. g_2) du nuage formé par le premier (resp. deuxième) sous-groupe
$$g_l = \frac{1}{n_{.l}} \sum_{\omega \in \Omega^a_l} Y(\omega)$$
3. les inerties intra-groupes de chaque sous-ensemble par rapport à leurs centre de gravité respectifs
$$I_l = \sum_{\omega \in \Omega^a_l} d^2(Y(\omega) - g_l)$$
4. l'inertie totale du nuage
$$I = \sum_{\omega \in \Omega^a} d^2(Y(\omega) - g)$$
5. et enfin, l'inertie inter-groupe par rapport au centre de gravité g
$$\begin{aligned} \delta(I_1, I_2) &= n_{.1} \times d^2(g_1, g) + n_{.2} \times d^2(g_2, g) \\ &= \frac{n_{.1} \times n_{.2}}{n_{.1} + n_{.2}} d^2(g_1, g_2) \end{aligned}$$

Quelques mesures

L'expression du critère dépend de la métrique utilisée. Nous donnons ici quelques formulations parmi lesquels nous distinguons différentes récritures d'indicateurs définis par ailleurs dans ce chapitre :

– pour le cas de l'indice de Gini, le gain d'entropie quadratique peut s'écrire [Lerman et Costa, 1996]

$$\Delta G(T_{Y/X}) = p_{.1}p_{.2}\sum_{k=1}^{K}(p_{k1} - p_{k2})^2$$

– avec une métrique du χ^2 [Bouroche et Tenenhaus, 1970], l'inertie inter-classes devient

$$\Delta\chi^2(T_{Y/X}) = p_{.1}p_{.2}\sum_{k=1}^{K}\frac{1}{p_{k.}}(p_{k1} - p_{k2})^2$$

3.5.2 Indices cosinus associés aux distances

Dans un article paru en 1992, [Fayyad et Irani, 1992] ont défini une classe de mesure, C-SEP, censée sélectionner les attributs tels que la partition induite "sépare autant que possible les individus de classes différentes, et garde ensemble les individus de la même classe". L'indicateur proposé est l'angle que font les deux vecteurs associés aux sommet enfants d'une bi-partition. L'indice cosinus résultant est défini par

$$\cos\theta(V_1, V_2) = \frac{V_1.V_2}{\|V_1\| \,.\, \|V_2\|} \tag{3.4}$$

où $V_l = (p_{1l}, \ldots, p_{Kl})^T$, le critère ORT que l'on devra maximiser s'écrit ainsi

$$ORT(T_{Y/X}) = 1 - \cos\theta(V_1, V_2)$$

En fait, nous pouvons associer à chaque indicateur de distance un indice cosinus, la mesure ORT de [Fayyad et Irani, 1992] n'est qu'un cas particulier dépendant de la mesure utilisée. On peut se poser la question d'ailleurs de la justesse de la formulation (équation 3.4) pour des vecteurs n'ayant pas la même origine. On trouve dans [Lerman et Costa, 1996] les formules correspondant aux distances de Gini et χ^2 :

$$\cos_G\theta(V_1, V_2) = \frac{\sum_{k=1}^{K}(p_{k1} - p_{k.})(p_{k2} - p_{k.})}{\sqrt{\left[\sum_{k=1}^{K}(p_{k1} - p_{k.})^2\right]\left[\sum_{k=1}^{K}(p_{k2} - p_{k.})^2\right]}} \tag{3.5}$$

$$\cos_{\chi^2}\theta(V_1, V_2) = \frac{\sum_{k=1}^{K}\frac{1}{p_{k.}}(p_{k1} - p_{k.})(p_{k2} - p_{k.})}{\sqrt{\left[\sum_{k=1}^{K}\frac{1}{p_{k.}}(p_{k1} - p_{k.})^2\right]\left[\sum_{k=1}^{K}\frac{1}{p_{k.}}(p_{k2} - p_{k.})^2\right]}} \tag{3.6}$$

Enfin, si l'on considère le nuage dual, c'est-à-dire K points dans $\mathbb{R}^2$, ces indices peuvent s'interpréter comme des coefficients de corrélation entre les deux sommets enfants [Lerman et Peter, 1985]. En effet, on observe (équations 3.5 et 3.6) que les indices cosinus se présentent sous la forme d'un rapport entre la covariance et le produit des variances respectives des deux nuages.

3.5.3 Autres distances

Distance de Kolmogorov Smirnov

Nous n'avons pas intégré dans les mesures ci-dessus la distance de Kolmogorov et Smirnov, très populaire dans la construction des arbres de décision binaires [Friedman, 1977] [Rounds, 1980], parce qu'elle revêt une forme assez particulière : elle est définie en principe uniquement pour les problèmes à deux classes avec des attributs prédictifs continus; certains auteurs lui accordent d'excellentes qualité en induction, notamment une meilleure résistance au bruits [Utgoff et Clouse, 1996] [Wehenkel, 1996].

Considérons un attribut continu $X(.)$, et un point de coupure a. Nous pouvons former deux blocs d'individus selon que $(X < a)$ ou $(X \geq a)$. Soient $F_{y_k}(X)$ les fonctions de répartitions conditionnellement aux classes $(k = 1, 2)$. La distance de Kolmogorov-Smirnov est définie comme suit

$$KS(T_{YX}) = \max_a |F_{y_1}(a) - F_{y_2}(a)| \tag{3.7}$$

La sélection des attributs sur un sommet se fait en choisissant la variable qui présente la plus grande valeur KS. Sous une hypothèse de matrice de coûts de mauvais classement symétrique, et une équirépartition a priori des classes, la valeur a^* qui maximise (3.7) est le point de coupure qui minimise l'espérance de mauvais classement bayésien [Friedman, 1977].

Dans la pratique, la construction des fonctions de répartitions empiriques $\widehat{F}_{y_l}$ suffisent pour calculer KS, elle est de plus non-paramétrique, nous pouvons utiliser sa distribution sous l'hypothèse d'égalité des fonctions de répartition pour tester la pertinence du découpage. Lorsque l'attribut prédictif est nominal, il faut construire deux groupes, par simple comptage on pourra définir les fonctions de répartitions empiriques, et donc qualifier les attributs les plus pertinents.

Enfin, pour les spécifications multiclasses $(K > 2)$, [Utgoff et Clouse, 1996] conseillent de former, pour chaque point de coupure candidat a, deux super-classes. La procédure à suivre est la suivante : trier les y_k selon les répartitions empiriques, construire deux groupes de classes adjacentes de manière à ce que leur distance soit maximale.

Distance de Mantaras

La distance de [Mantaras, 1991] est encore plus particulière que celle de Kolmogorov-Smirnov car elle mesure la distance entre la distribution de probabilités des classes et la distribution de probabilités des modalités de l'attribut prédictif. De fait, à l'inverse des précédentes, elle s'applique parfaitement dans le cas des attributs comportant plus de deux modalités.

Son estimateur est le suivant

$$\widehat{DM}(T_{Y/X}) = 1 - \frac{\Delta\widehat{S}(T_{Y/X})}{\sum_{l=1}^{L}\sum_{k=1}^{K}\frac{n_{kl}}{n}\log(\frac{n_{kl}}{n})} \tag{3.8}$$

L'interprétation de ce critère devient aisée lorsque l'on [Wehenkel, 1996] s'aperçoit qu'on peut le reconstruire en utilisant un autre gain d'entropie normalisé qui lui s'apparente à un coefficient de corrélation [Kvalseth, 1987]. La distance de Mantaras prend les mêmes valeurs que ce dernier (1 quand la corrélation est maximale, tendant vers 0 quand la corrélation est faible).

[Mantaras, 1991] affirme que son critère n'est pas biaisé en faveur des attributs multivalués en utilisant une démonstration formelle issue d'une spécification de [Quinlan, 1988a]. Nous essaierons plus loin de vérifier autrement cette assertion.

3.5.4 Problème des partitions non binaires

Mis à part la distance de Mantaras, toutes les autres mesures sont définies pour des partitions binaires. Force est de constater que ce n'est pas toujours approprié. De fait, nous sommes obligés de procéder à des regroupements ou à des transformations de variables :

- la première stratégie consiste à effectuer un codage disjonctif complet sur les variables, cela ne pose aucun problème pratique si ce n'est la perte de l'information sur les relations d'exclusion entre les modalités d'une variable (un individu ne peut pas être en même temps masculin et féminin);

- le seconde consiste à choisir une modalité contre les autres. En les testant toutes, nous pouvons choisir la bipartition qui maximise la mesure [Munteanu, 1996]. On reproche à cette méthode de trop fragmenter les données, il se peut très bien que plusieurs des modalités de l'attribut prédictif soient de la même manière liées à une classe (par exemple dans une enquête, les "très satisfaits" et "satisfaits" concernant l'utilisation d'un dentifrice, peuvent présenter un comportement d'achat identique);

- la troisième, celle qui a reçu le plus de suffrages [Cheng *et al.*, 1988] [Cestnik *et al.*, 1987a], revient à construire un attribut binaire regroupant les différentes modalités. Si les arguments ne manquent pas pour une telle stratégie, nous l'étudierons en profondeur dans le chapitre sur les stratégies de construction des graphes, elle pose un problème de complexité combinatoire ($2^{L-1} - 1$ partitions possibles) qui la rend impraticable dès que le nombre de modalités augmente. Différents travaux se sont penchés activement sur sa résolution, sous certaines hypothèses restrictives, il est possible de ramener la recherche de la solution parmi $L - 1$ partitions judicieusement choisies [Breiman *et al.*, 1984] [Asseraf, 1996].

3.6 Mesures fondées sur la causalité

Ces indicateurs tentent d'exprimer l'association entre deux variables catégorielles nominales [Agresti, 1990]. Il en existe un grand nombre, toutes répondent parfaitement aux trois premières

propriétés édictées dans la section 3.3.2. Nous en verrons essentiellement deux variétés :

- les mesures d'écarts à l'indépendance basées sur le χ^2 de Pearson
- les mesures non-symétriques de causalité, parmi lesquelles nous retrouverons bon nombre des indicateurs précédemment décrits.

Ces mesures ayant été très utilisées en statistique, notamment dans l'étude des relations de causalité et des réseaux causaux [Olszak et Ritschard, 1995], elles sont pour la plupart associées à des distributions de probabilités sous certaines hypothèses. Nous pourrons à loisir tester l'absence ou la présence d'une association entre la classe et l'attribut prédictif. Une telle procédure peut d'ailleurs constituer une règle d'arrêt dans la construction des graphes d'induction.

3.6.1 Mesures d'écart à l'indépendance du χ^2

Le test le plus connu pour apprécier l'indépendance dans les tableaux de contingence a été mis au point par Pearson (1904). La statistique du test s'écrit

$$\chi^2 = \sum_{k=1}^{K} \sum_{l=1}^{L} \frac{n(n_{kl} - \frac{n_{k.}n_{.l}}{n})^2}{n_{k.}n_{.l}}$$

Cet indicateur mesure en fait l'écart entre le tableau actuel et le tableau représentant l'indépendance entre les deux attributs, dont les cases n_{kl} seraient remplies par le produit des marges.

Sous l'hypothèse nulle d'indépendance entre les attributs, χ^2 suit une loi de χ^2 à $(K-1)(L-1)$ degrés de liberté, et permet ainsi de construire un test rejetant cette hypothèse lorsque χ^2 est suffisamment grand. Il existe quand même quelques restrictions quant à l'utilisation de ce test, [Siegel, 1956] par exemple montre son inefficience lorsque 20% des effectifs estimés sous l'hypothèse d'indépendance sont inférieurs à 5, ou encore lorsqu'il y a au moins une case inférieure à la valeur 1. Si une de ces deux conditions est remplie, on sait que ce test conclut trop souvent en faveur de l'hypothèse alternative.

Deux versions normalisées de χ^2 présentent le double avantage de varier entre des bornes connues (0 indépendance parfaite, 1 liaison maximale) et de tenir compte explicitement de la complexité du découpage. Ces indicateurs ont d'ailleurs été utilisés par [Lechevallier, 1990] en discrétisation pour comparer des découpages en nombres d'intervalles différents. Nous aborderons plus en détail ce problème plus loin. Ces mesures sont le t de Tschuprow [Tschuprow, 1921] et le v de Cramer [Cramer, 1946] dont voici les formules respectives :

$$t = \frac{\chi^2}{n\sqrt{(K-1)(L-1)}}$$

$$v = \sqrt{\frac{\chi^2}{n \ \min\{K-1, L-1\}}}$$

Bien entendu, comme nous avons pu le constater tout le long de ce chapitre, il existe plusieurs interprétations de ces mesures, le t de Tschuprow peut par exemple être assimilé au cosinus de l'angle que font deux distributions de classes [Saporta, 1975].

Il reste néanmoins qu'il s'agit bien d'écart à l'indépendance et non de causalité ici, la transposition du tableau de contingence ne change en rien la valeur de l'indicateur [Olszak, 1995].

3.6.2 Causalité à l'aide de mesures non-symétriques

Parmi les indicateurs non-symétriques, caractérisant l'influence de la connaissance d'un attribut $X(.)$ sur la prédiction des valeurs prises par une autre variable $Y(.)$, figurent les mesures PRE (Proportional reduction error) initiées par [Goodman et Kruskall, 1954]. Elles sont d'une interprétation simple : elles estiment la réduction normalisée de la probabilité de mauvaise prédiction de la valeur de $Y(.)$ avant (e_1) et après (e_2) connaissance de $X(.)$ (équation 3.9).

$$\frac{e_1 - e_2}{e_1} \tag{3.9}$$

La mesure λ_{YX}

L'indicateur λ_{YX} [Goodman et Kruskall, 1954] est calculé de la manière suivante :

- si l'on adopte le choix de la classe $y_{k'}$ la plus fréquente a priori, la probabilité d'erreur s'écrit

$$e_1 = 1 - p_{k'.}$$

- pour une modalité x_l de $X(.)$, la probabilité de bien classer si l'on choisit la conclusion y_{k_l} est $p_{k_l,l}$. En moyenne donc, la probabilité de bien classer connaissant la valeur prise par $X(.)$ est égale à $\sum_{l=1}^{L} p.l \times p_{k_1,l}$, l'erreur associée

$$e_2 = 1 - \sum_{l=1}^{L} p_{.l} \times p_{k_1,l}$$

La mesure λ_{YX} de réduction d'erreur proportionnelle s'écrit alors

$$\lambda_{YX} = \frac{e_1 - e_2}{e_1}$$

C'est un indicateur normalisé, et non-symétrique. Si on transpose le tableau de contingence T, nous vérifions aisément que $\lambda_{YX} \neq \lambda_{XY}$.

En passant aux estimations du maximum de vraisemblance des probabilités :

$$\widehat{\lambda}_{YX} = \frac{(1 - \frac{n_{k'.}}{n}) - (1 - \sum_{l=1}^{L} \frac{n_{k_1,l}}{n})}{1 - \sum_{k' \neq k} \frac{n_{k.}}{n}}$$

Le principal reproche qu'on lui adresse est qu'il nous oblige à définir une conclusion sur toutes les colonnes du tableau. Cette situation n'est pas sans rappeler les faiblesses du taux d'erreur

en induction. Si nous revenons à l'exemple de [Utgoff et Clouse, 1996] (cf. 3.3.1), dans la marge droite et sur toutes les colonnes, nous conclurons toujours à la classe y_2, et la mesure de causalité calculée est nulle :

$$\begin{aligned} \widehat{\lambda}_{YX} &= \frac{0.0099 - (1 - 0.9901)}{0.0099} \\ &= 0 \end{aligned}$$

Il est nécessaire de tenir compte de la structure de l'erreur.

La mesure τ_{YX}

[Goodman et Kruskall, 1954] ont mis au point une deuxième mesure plus riche, qui s'intéresse cette fois à la structure de l'erreur pour calculer la réduction proportionnelle (équation 3.9).

Si on s'intéresse à la distribution a priori de $Y(.)$, la probabilité qu'un individu porte la valeur y_k est égale à $p_{k.}$. Si nous adoptons une stratégie d'affectation proportionnelle à la distribution des classes, i.e qui consiste à conclure à la classe y_k avec la probabilité $p_{k.}$, la probabilité de conclure à juste titre à la classe y_k est égale à $p_{k.} \times p_{k.}$, et la probabilité de conclure à tort s'écrit $p_{k.} \times (1 - p_{k.})$.

Dès lors, pour l'ensemble des K classes, la probabilité d'erreur s'écrit

$$e_1 = 1 - \sum_{k=1}^{K} p_{k.}^2$$

$$e_1 = \sum_{k=1}^{K} p_{k.}(1 - p_{k.}) \tag{3.10}$$

La formule (3.10) n'est pas sans rappeler l'indice de Gini, connue encore sous l'appellation entropie quadratique. L'expression de l'erreur moyenne conditionnelle est obtenue avec le même raisonnement

$$e_2 = \sum_{l=1}^{L} p_{.l} \sum_{k=1}^{K} p_{kl}(1 - p_{kl})$$

Si nous appliquons la forme générale des mesures P.R.E (équation 3.9), nous nous rendons compte que la mesure τ_{YX} de [Goodman et Kruskall, 1954] n'est rien d'autre qu'un gain normalisé d'entropie quadratique. En procédant de manière similaire, [Theil, 1970] a introduit la mesure d'association u_{YX} qui revient à calculer un gain normalisé d'entropie de Shannon :

$$u_{YX} = \frac{S(T_Y) - S(T_{Y/X})}{S(T_Y)}$$

3.6.3 Analogie avec les variables numériques - Interprétation en terme de corrélation

Historiquement, l'étude des mesures d'association a été initiée par des statisticiens qui se sont intéressés à l'association entre variables catégorielles après que les principaux résultats sur les relations entre variables numériques aient été bien établis [Olszak, 1995]. A cette ascendance, nous voyons deux avantages :

- il existe plusieurs résultats très intéressants sur la distribution des statistiques empiriques ($\widehat{\lambda}_{YX}$, $\widehat{\tau}_{YX}$, $\widehat{u}_{YX}$) des mesures d'association, utilisant les estimateurs du maximum de vraisemblance des probabilités. Ainsi, on a pu prouver leur normalité asymptotique, les variances ont été chiffrées en employant la "méthode delta" [Goodman et Kruskal, 1972]. De fait, nous pouvons à la fois construire un intervalle de confiance autour des valeurs prises par les statistiques calculées, et procéder à des tests de significativité de l'association.

- l'analogie avec les résultats sur les variables numériques ont permis de dégager une interprétation assez séduisante en terme de corrélation qui jette un pont entre les deux approches. [Wehenkel, 1996] recense les différentes formes de normalisation du gain d'entropie, leurs relations et leur interprétation en terme de corrélation. Dans ce dernier cas, puisque corrélation n'est pas synonyme de causalité, du moins en ce qui concerne les variables numériques, on opère une transformation de manière à rendre la mesure symétrique [Zhou et Dillon, 1991] [Kvalseth, 1987]. On notera qu'une telle interprétation n'est pas propre à l'induction de règles à l'ordre O^+, on peut la retrouver également dans certaines méthodes de Programmation Logique Inductive [Furnkranz, 1994].

3.7 Mesures fondées sur les méthodes de comparaisons par paires

3.7.1 Principe

Cette famille de mesures est définie sur les relations que les deux variables $Y(.)$ et $X(.)$ à comparer induisent sur l'ensemble d'apprentissage Ω^a [Lerman et Costa, 1996]. Nous pouvons assimiler chacune des variables comme une application de Ω^a vers un ensemble de catégories : les modalités de la variable.

Soit $M_Y = \{y_1, \ldots, y_K\}$ (resp. $M_X = \{x_1, \ldots, x_L\}$) l'ensemble des modalités prises par la variable $Y(.)$ (resp. $X(.)$). Chaque modalité des variables $X(.)$ et $Y(.)$ permet de circonscrire des partitions de l'échantillon d'apprentissage Ω^a :

$$\begin{aligned} \Omega^a(y_k) &= \{\omega \in \Omega^a \,/\, Y(\omega) = y_k\} \\ \Omega^a(x_l) &= \{\omega \in \Omega^a \,/\, X(\omega) = x_l\} \end{aligned}$$

Pour chaque variable, nous disposons d'un ensemble de partitions

$$\begin{aligned} \Omega^a(Y) &= k \cup \Omega^a(y_k) \\ \Omega^a(X) &= l \cup \Omega^a(x_l) \end{aligned}$$

Le critère "brut" d'association de Lerman se déduit par le comptage de paires d'objets réunis par les deux partitions

$$\begin{aligned} s(Y,X) &= card[\Omega^a(Y) \cap \Omega^a(X)] \\ s(Y,X) &= \sum_{k,l} \frac{n_{kl}(n_{kl}-1)}{2} \end{aligned}$$

où $n_{kl} = card[\Omega^a(y_k) \cap \Omega^a(x_l)]$

3.7.2 Mesures corrigées

A partir de cette première mesure, [Lerman, 1992a] a défini un ensemble de mesures corrigées. Le critère $Q_1(Y,X)$ par exemple résulte du centrage réduction de $s(Y,X)$ par son espérance et sa variance *dans une hypothèse d'indépendance des variables.* Ce procédé n'est pas sans rappeler les méthodes de construction de statistiques de test de Neymann-Pearson, l'indicateur peut alors s'interpréter comme un écart à l'indépendance.

Nous décrivons ici les formules idoines [Lerman et Costa, 1996]:

$$\begin{aligned} E[s_{H_0}(Y,X)] &= \lambda\mu \\ V[s_{H_0}(Y,X)] &= \lambda\mu + \rho\sigma + \theta\xi - \lambda^2\mu^2 \end{aligned}$$

où

$$\begin{aligned} \lambda &= \sum_{k=1}^{K} \frac{n_{k.}(n_{k.}-1)}{\sqrt{2n(n-1)}} \\ \rho &= \sum_{k=1}^{K} \frac{n_{k.}(n_{k.}-1)(n_{k.}-2)}{\sqrt{n(n-1)(n-2)}} \\ \mu &= \sum_{l=1}^{L} \frac{n_{.l}(n_{.l}-1)}{\sqrt{2n(n-1)}} \\ \sigma &= \sum_{l=1}^{L} \frac{n_{.l}(n_{.l}-1)(n_{.l}-2)}{\sqrt{n(n-1)(n-2)}} \\ \xi &= \frac{[\sum_{l=1}^{L} n_{.l}(n_{.l}-1)]^2 - 2[\sum_{l=1}^{L} n_{.l}(n_{.l}-1)(2n_{.l}-3)]}{2\sqrt{n(n-1)(n-2)(n-3)}} \end{aligned}$$

Le critère centré réduit s'écrit alors

$$Q_1(Y,X) = \frac{s(Y,X) - E[s_{H_0}(Y,X)]}{\sqrt{V[s_{H_0}(Y,X)]}}$$

Cette famille de mesures est encore très peu utilisée dans la construction des arbres de décision, on peut se demander quel intérêt y a-t-il de produire de nouveaux indicateurs qui finalement en performances de classement ne se comportent pas mieux que les mesures couramment utilisées [Breiman *et al.*, 1984] [Quinlan, 1986b] [Kass, 1980]. Une première réponse a été suggérée dans [Lerman et Costa, 1996], ils rejoignent par là le point de vue de nombreux chercheurs [Buntine, 1990] [Schaffer, 1993a] : le mythe du prédicteur universel n'existe pas, la meilleure attitude est d'essayer l'outil le plus approprié compte tenu de nos connaissances sur l'étude à réaliser. Dans le cas de ces mesures, elles ont été élaborées dans le cadre particulier de la binarisation d'attributs prédictifs comportant de très nombreuses modalités, et où certaines classes sont rares. Mais au final, les études empiriques sur l'identification de la structure secondaire d'une protéine ont montré que les mesures fondées sur les comparaisons par paires, au moins dans la construction d'arbres binaires, donc avec une influence nulle de l'éventuel biais en faveur des attributs multivalués, se comportent ni mieux ni moins bien que les critères usuels de construction d'arbre de décision.

3.8 La pénalisation des partitions trop fragmentaires dans le découpage n-aire

Plusieurs études ont montré que le gain informationnel de Shannon (cf. équation 3.1) favorisait éxagérément les attributs prédictifs comportant beaucoup de modalités lors du partitionnement des individus [Hart, 1984] [White et Liu, 1994]. La conséquence en est la fragmentation des données qui aboutit à des règles d'affectation peu fiables, et un graphe surdimensionné.

Considérons un exemple simple pour mieux appréhender ce problème (figure 3.4). Nous sommes en présence de deux arbres de décision à un niveau induisant deux partitions pures : la première (figure 3.4.a) avec deux sommets enfants, la seconde en quatre sous-ensembles (figure 3.4.b). Intuitivement, nous préférerons le premier arbre parce qu'il est tout aussi précis que le second, mais avec des règles qui couvrent plus d'individus, donc d'une meilleure fiabilité en généralisation[11]. Le problème qui nous préoccupe ici est de trouver une mesure qui nous permette à coup sûr de choisir la première solution. Dans ce qui suit, nous présentons quelques solutions répondant à des philosophies différentes. Notons qu'avec la binarisation des attributs [Cestnik *et al.*, 1987a], i.e le regroupement en deux paquets des modalités de l'attribut prédictif, de tels soucis n'ont plus lieu d'être.

11. Dans la chapitre sur les règles, nous donnerons une définition plus formelle de cette intuition "à précision égale, généralité plus grande".

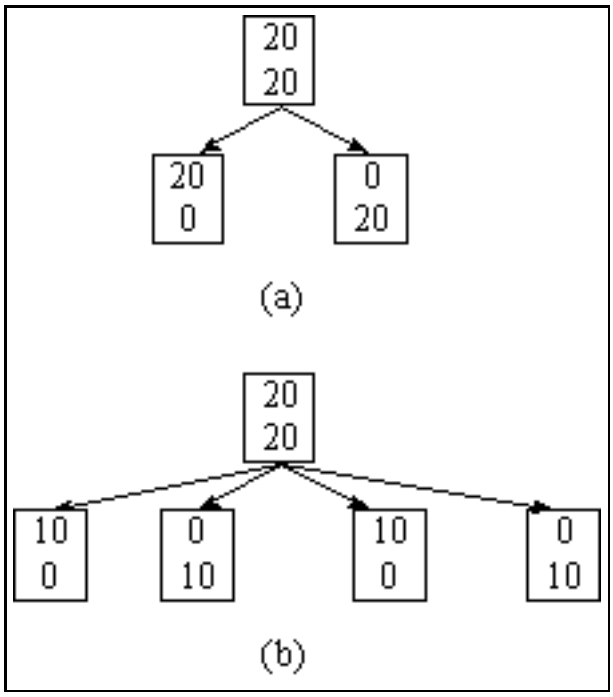

FIG. 3.4 – *Deux partitions alternatives sur un noeud*

3.8.1 Pénalisation des sommets à effectifs trop faibles

Afin de favoriser les regroupements lors de la discrétisation des attributs continus, [Zighed et Rakotomalala, 1996a] ont introduit une nouvelle mesure inspirée des travaux sur les mesures d'incertitude de [Zighed *et al.*, 1992]. La nouvelle formulation rajoute un deuxième terme qui pénalise les groupes de trop faibles effectifs, elle s'écrit comme suit

$$\varphi(T_{Y/X}) = \sum_{l=1}^{L} \alpha \frac{n_{.l}}{n} \sum_{k=1}^{K} \frac{n_{kl} + \lambda}{n_{.l} + K\lambda}(1 - \frac{n_{kl} + \lambda}{n_{.l} + K\lambda}) + (1 - \alpha)\frac{\lambda K}{n_{.l}}$$

avec $0 < \alpha < 1$.

[Rabaseda, 1996] a vérifié que cette mesure remplissait bien les propriétés présentées dans la section 3.3.2. Le choix du paramètre α peut se faire de différentes manières : par choix d'expert, selon la complexité désirée du graphe, on choisira une valeur élevée (graphe plus grand) ou faible (graphe très réduit); par cross-validation, on fixe une série de valeurs α_i, et on choisit celle qui minimise l'erreur; enfin, et c'est ce que nous présentons ici, en fixant une contrainte d'effectif minimum sur un sommet.

Afin d'assurer une certaine fiabilité au classifieur, il est d'usage de fixer une taille minimale aux sommets que l'on veut construire [Quinlan, 1993a]. Les graphes sont moins dépendants de cette obligation, lorsqu'un sommet de faible taille apparaît, il suffit de le fusionner avec un autre sommet de plus proche distribution. En fixant une valeur adéquate de α, nous pouvons intégrer explicitement cette contrainte dans la construction de graphe, les sommets de taille inférieure à

n^* seront fermement enjoints au regroupement avec les autres, même si la distribution des classes en est assez éloignée.

Nous avons utilisé la méthode générique suivante pour fixer la valeur α dans la pratique. Soit deux sommets d'un graphe de taille inférieure à la taille limite n^* que l'on s'est fixé (afin d'alléger la notation nous poserons $n^* = n$)

$Y = y_1$		$n-1$		0	
$\vdots$	...	0	...	1	...
$Y = y_k$		$\vdots$		0	
$\vdots$		$\vdots$		$\vdots$	
$Y = y_K$		0		0	

On provoquant leur fusion, le sommet construit remplira la contrainte de taille même si par ailleurs les distributions de classes diffèrent

$Y = y_1$		$n-1$	
$\vdots$	...	1	...
$Y = y_k$		$\vdots$	
$\vdots$		$\vdots$	
$Y = y_K$		0	

Afin d'aboutir au regroupement, nous devons fixer une valeur de α^* adéquate qui, en quelque sorte, élargit la propriété de fusion en intégrant la sensibilité à une taille limité d'effectif. Si nous calculons les incertitudes locales des deux situations ci-dessus, nous obtenons respecivement :

$$\begin{aligned} e_1 &= \alpha\frac{n-1}{n}[\frac{n-1+\lambda}{n+K\lambda}(1-\frac{n-1+\lambda}{n+K\lambda})+(K-1)\frac{\lambda}{n+K\lambda}(1-\frac{\lambda}{n+K\lambda})] && (3.11)\\ &\quad +(1-\alpha)\frac{\lambda K}{n-1} && (3.12)\\ &\quad +\alpha\frac{1}{n}[\frac{1+\lambda}{n+K\lambda}(1-\frac{1+\lambda}{n+K\lambda})+(K-1)\frac{\lambda}{n+K\lambda}(1-\frac{\lambda}{n+K\lambda})] && (3.13)\\ &\quad +(1-\alpha)\frac{\lambda K}{1} && (3.14)\end{aligned}$$

et

$$\begin{aligned} e_2 &= \alpha\frac{n-1}{n}[\frac{n-1+\lambda}{n+K\lambda}(1-\frac{n-1+\lambda}{n+K\lambda}) && (3.15)\\ &\quad +\frac{1+\lambda}{n+K\lambda}(1-\frac{1+\lambda}{n+K\lambda})+(K-2)\frac{\lambda}{n+K\lambda}(1-\frac{\lambda}{n+K\lambda})] && (3.16)\\ &\quad +(1-\alpha)\frac{\lambda K}{n} && (3.17)\end{aligned}$$

La solution α^* est telle que

$$e_1 - e_2 = 0 \tag{3.18}$$

qui peut être obtenu littéralement puisque l'équation 3.18 est linéaire en α.

3.8.2 Pénalisation de la complexité à l'aide d'une formulation bayésienne

Depuis le début des années 90, la théorie bayésienne en induction a connu un développement fulgurant avec les travaux de [Buntine, 1991] et [Wehenkel, 1990]. Dans le même temps, se refusant à produire une estimation toujours délicate des probabilités a priori, d'autres chercheurs se sont penchés sur des approches similaires qui sont la description minimale des données [Rissanen, 1978] et la longueur minimale des messages [Wallace et Freeman, 1987], dont d'ailleurs [Kononenko, 1995] affirme qu'elles sont les moins biaisées en faveur des attributs multivalués.

Nous relatons ici les résultats issus d'un papier qui ne fut jamais publié de [Wehenkel, 1992], dont les prémices étaient déjà explicitées dans [Wehenkel, 1990]. Il utilise la théorie bayesienne tout en donnant à ses résultats une interprétation en terme de théorie de la description des messages. Dans la section 3.4.2, nous avions montré que le gain d'entropie de Shannon revenait à maximiser la vraisemblance, et qu'en réalité il s'agissait là d'une fraction seulement d'une formulation plus globale. Ce constat est commun à toutes les approches citées ci-dessus, tout le problème revient alors à en décrire la teneur.

Dans la théorie bayesienne, nous cherchons le modèle M le plus probable compte tenu des données. Nous cherchons à maximiser l'expression

$$P(M/\Omega^a) = \frac{P(\Omega^a/M) \times P(M)}{P(\Omega^a)} \tag{3.19}$$

Quel que soit le modèle choisi, l'expression au dénominateur reste constante. En passant au logarithme à base 2 et en multipliant par -1, nous devons donc minimiser

$$L(\Omega^a, M) = -\log P(\Omega^a/M) - \log P(M) \tag{3.20}$$

L'auteur montre que le premier terme du deuxième membre de l'équation 3.20 est à un coefficient multiplicatif n près l'entropie de Shannon, reste à évaluer le second terme $-\log P(M)$.

Sans entrer dans le débat pour ou contre le bayesianisme, il est évident que le choix de la distribution de probabilité $P(M)$ conditionne complètement l'action de la mesure sur la construction du graphe. Une justification possible serait d'instaurer un biais de préférence en faveur des modèles les moins complexes afin notamment d'éviter le sur-apprentissage face aux données bruitées [Schaffer, 1993a]. Dans cette optique, [Wehenkel, 1992] propose que la complexité $C(M)$ d'un arbre de décision soit quantifiée à l'aide du nombre de feuilles qu'elle comporte, les modèles de même complexité sont équiprobables a priori. Au moyen du critère du maximum de l'entropie,

l'auteur dérive une distribution exponentielle de paramètre a positif, que l'on pourra faire varier selon le biais désiré

$$P(M) \propto e^{-a \times C(M)}$$

L'équation 3.20 s'interprête comme maintenant la longueur de représentation

$$L(\Omega^a, M) = n \times S(T_{Y/X}) + a \times C(M)$$

Afin d'apprécier la qualité globale du modèle, nous devons confronter le coût de description de tous les individus, l'entropie sur le sommet initial de l'arbre, et la longueur de représentation du modèle

$$Q(\Omega^a, M) = n \times S(T_Y) - [n \times S(T_{Y/X}) + a \times C(M)]$$

$$Q(\Omega^a, M) = n \times \Delta S(T_{Y/X}) - a \times C(M) \tag{3.21}$$

La mesure étant additive, pour un éclatement sur un sommet, nous voyons bien qu'un attribut multivalué sera plus pénalisé qu'un attribut ayant peu de modalités par le terme $a \times C(M)$. Cette pénalité sera d'autant plus forte que le paramètre a sera élevé. Dans la pratique, l'auteur propose de l'optimiser par cross-validation [Wehenkel, 1993].

Nous observons également qu'une règle d'arrêt assez naturelle de l'expansion de l'arbre peut être dérivée de $Q(\Omega^a, M)$, nous cherchons la partition qui maximise cette quantité. Ceci est possible parce que $Q(\Omega^a, M)$ peut être négative lorsque le nombre de feuilles est éxagérément élevé.

Nous remarquons que cette formulation bayésienne de [Wehenkel, 1993] revient tout simplement à introduire une pénalité fonction du nombre de modalités de l'attribut prédictif introduit dans le découpage. Cette idée n'est pas nouvelle, les mesures t de Tschuprow et v de Cramer dérivées du χ^2 l'introduisent explicitement au dénominateur de l'indicateur et deviennent des fonctions décroissantes du nombre de modalités de la variable prédictive, toutes choses égales par ailleurs. Ainsi, on contrebalance le biais en faveur des attributs multivalués. [Rauber et Steiger-Garccedillo, 1993] utilise avec succès ces mesures dans la construction d'arbres de décision et ramène l'étude de ces derniers à l'analyse des évolutions d'un tableau de contingence.

3.8.3 Formulations statistiques : utilisation de la loi de distribution du χ^2

Tous les critères étant estimés à partir d'un échantillon Ω^a issu de la population parente Ω, nous pouvons en dériver une distribution statistique et vérifier l'existence d'une liaison significativement forte, hors fluctuations d'échantillonnage, entre la variable à prédire $Y(.)$ et l'attribut $X(.)$.

On construit un test qui s'écrit généralement :

$$\begin{aligned} H_0 &: X(.) \textit{ et } Y(.) \textit{ sont indépendants} \\ H_1 &: X(.) \textit{ est lié avec } Y(.) \end{aligned}$$

Une mesure quelconque $\Psi(T)$ peut devenir la statistique du test de Neymann et Pearson, pour peu que l'on arrive à en expliciter la distribution de probabilité sous l'hypothèse d'indépendance entre $X(.)$ et $Y(.)$. La plupart du temps, puisque l'on utilise une estimation du maximum de vraisemblance de la probabilité qui est asymptotiquement normale, tout forme quadratique qui en est issue suit une loi du χ^2.

Si nous reprenons le gain d'entropie de Shannon, nous pouvons en construire la statistique G qui fût utilisée par [Mingers, 1987][12] [Wehenkel, 1990] dans les arbres de décision :

$$G(T_{Y/X}) = 2 \times n \times \ln(2) \times \Delta\widehat{S}(T_{Y/X})$$

Elle suit une loi du χ^2 à $(K-1)(L-1)$ degrés de liberté sous l'hypothèse nulle. La région critique du test, où l'on décide que l'éclatement est significativement pertinent, s'écrit

$$G(T_{Y/X}) \geq \chi^2_{1-\alpha}(K-1)(L-1)$$

avec α le risque de première espèce, i.e la probabilité de conclure à tort à la pertinence du découpage, associé au test.

A l'instar de tous les tests, cette région critique peut se traduire en

$$\Pr[\chi^2 \geq G(T_{Y/X})] \leq \alpha \tag{3.22}$$

On peut dès lors utiliser le risque critique du test (le premier membre de l'équation 3.22) pour qualifier une mesure. Son interprétation est relativement aisée, elle revient à considérer l'attribut prédictif qui induit le découpage tel que l'on s'éloigne le plus de la situation d'indépendance. La mesure que l'on essaiera de maximiser lors de la sélection des attributs prendra la forme

$$\Psi(T_{Y/X}) = 1 - \Pr[\chi^2 \geq G(T_{Y/X})]$$

qui toutes choses égales par ailleurs (n, $\Delta\widehat{S}(T_{Y/X})$ et K), est une fonction décroissante de L.

On remarque qu'à la différence des corrections précédentes, il n'y a pas de paramètre à fixer dans ce type de mesure. Cela peut être à la fois un avantage, parce qu'en l'absence de connaissances a priori l'exploration se fait automatiquement, et un inconvénient, parce que l'expert ne dispose plus d'un dispositif de pilotage pour guider l'exploration des solutions.

12. Mingers a utilisé une spécification erronée de cette statistique dans ses travaux [White et Liu, 1994].

3.8.4 Evaluation du biais en faveur des attributs multivalués à travers différentes expérimentations

Nous disposons de deux manières différentes pour vérifier si une mesure de qualité de partition n'est pas biaisée en faveur des attributs multi-valués.

Evaluation empirique

La première consiste à opérer une expérimentation qui consiste à générer des données synthétiques et vérifier statistiquement l'influence du nombre de modalités de l'attribut sur la mesure [White et Liu, 1994] [Kononenko, 1995]. Cette méthode est la plus simple, pour peu que le protocole expérimental soit bien construit, les résultats ne souffrent pas de contestation. Il y est apparu notamment que les mesures corrigées du χ^2 ou fondées sur la description minimale des messages étaient effectivement non biaisées, à l'inverse du gain ratio et de la distance de Mantaras.

Nous pouvons également prendre des bases de données réelles, les fameux "benchmark" disponibles sur les serveurs Internet, et analyser les résultats des différentes méthodes : les mesures qui favorisent les attributs multivalués conduisent généralement à la construction d'arbres "larges", contenant beaucoup de feuilles [Ho et Nguyen, 1996].

Evaluation analytique

La seconde évaluation consiste à déterminer des propriétés analytiques qui permettent de qualifier l'influence du nombre de modalités sur la mesure : si elle est positive, nous pouvons affirmer que l'indicateur n'est pas approprié. Toute la problématique réside alors dans la justification de la formulation choisie. Dans cet ordre d'idée, [Quinlan, 1988a] propose l'évaluation suivante : soit un attribut $X(.)$ prenant ses valeurs dans $\{x_1, \ldots, x_i, \ldots, x_L\}$, soit un attribut $X'(.)$ formé à partir de $X(.)$ et dont on a subdivisé aléatoirement la valeur x_i en x_{i_1} et x_{i_2}. La mesure est biaisée en faveur des attributs multi-valués si

$$H(T_{Y/X'}) \geq H(T_{Y/X}) \tag{3.23}$$

En utilisant ce type de spécification, [Mantaras, 1991] et [Quinlan, 1986b] ont montré que leurs indicateurs ne favorisaient pas les attributs multi-valués, ce qui a été réfuté empiriquement par [White et Liu, 1994]. Face à ces contradictions, nous pensons que la spécification analytique de l'évaluation ci-dessus n'est peut-être pas justifiée.

[Quinlan, 1986b] lui-même reconnaît que dans le cas extrême où les distributions de probabilités des classes sont identiques i.e $T_{Y/(X'=x_{i_1})} = m \times T_{Y/(X'=x_{i_2})}$, on observe l'égalité dans l'équation 3.23. Nous pensons que pour vérifier la présence du biais face aux attributs multivalués, nous devons utiliser une spécification plus exigeante, ce qui est le cas avec la propriété de **fusion.**

Elle permet d'appréhender la capacité de l'indicateur à choisir les concepts les moins complexes, elle est en effet fondée sur le principe suivant : si deux colonnes du tableau de contingence T possèdent une structure de distribution de classes identiques, leur réunion engendrera une diminution de la mesure indiquant un raffermissement de l'association entre l'attribut prédictif et la variable à prédire. L'idée sous-jacente étant de spécifier à précision égale, la disposition la moins complexe (en nombre de colonnes) ou constituée de sous-groupes de plus forts effectifs. Nous avons d'ailleurs vérifié que toutes les mesures de la section 3.8 remplissaient effectivement cette propriété.

En revanche, nous montrons ici que la distance de Mantaras ainsi que le gain ratio ne remplissent pas cette propriété. Rappelons pour mémoire que ce sont deux gains d'information de Shannon normalisés. La première par l'entropie conjointe de $S(T_{YX})$ (voir équation 3.8) et la seconde par la quantité de bits pour connaître la modalité prise par $X(.)$

$$gain\_ratio = \frac{\Delta \widehat{S}(T_{Y/X})}{-\sum_{l=1}^{L} \frac{n_{.l}}{n} \log(\frac{n_{.l}}{n})}$$

Preuve :

Il suffit de montrer que le gain informationnel est invariant au regard de la propriété 5, et nous pouvons conclure que ces mesures ne pénalisent pas les partitions de complexité élevée. La propriété s'énonce comme suit : "Soit $T = (T_1|\ldots|T_i|\ldots|T_j|\ldots|T_L)$ un tableau de contingence. S'il existe un doublet i et j avec $T_i = s \times T_j$, alors la partition fusionnée $T'' = (T_1|\ldots|T_i + T_j|\ldots|T_L)$ est telle que $\Psi(T'') < \Psi(T)$"

Pour le calcul du gain, les distributions marginales T_Y et T''_Y sont identiques, de fait $\widehat{S}(T_Y) = S(T''_Y)$. Vérifions ce qu'il en est pour l'entropie conditionnelle.

$$\begin{aligned}
\widehat{S}(T_{Y/X}) &= \sum_{l=1}^{L} \frac{n_{.l}}{n} \sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}} \log(\frac{n_{kl}}{n_{.l}}) \\
&= \cdots + \frac{n_{.i}}{n} \sum_{k=1}^{K} \frac{n_{ki}}{n_{.i}} \log(\frac{n_{ki}}{n_{.i}}) + \cdots + \frac{n_{.j}}{n} \sum_{k=1}^{K} \frac{n_{kj}}{n_{.j}} \log(\frac{n_{kj}}{n_{.j}}) + \cdots
\end{aligned}$$

Puisque $n_{ki} = m \times n_{kj}$ $(k = 1, ..., K)$

$$\begin{aligned}
\sum_{k=1}^{K} \frac{n_{ki}}{n_{.i}} \log(\frac{n_{ki}}{n_{.i}}) &= \sum_{k=1}^{K} \frac{m \times n_{ki}}{m \times n_{.i}} \log(\frac{m \times n_{ki}}{m \times n_{.i}}) \\
&= \sum_{k=1}^{K} \frac{(m+1) \times n_{kj}}{(m+1) \times n_{.j}} \log(\frac{(m+1) \times n_{kj}}{(m+1) \times n_{.j}}) \\
&= \sum_{k=1}^{K} \frac{n_{ki} + n_{kj}}{n_{.i} + n_{.j}} \log(\frac{n_{ki} + n_{kj}}{n_{.i} + n_{.j}})
\end{aligned}$$

Ainsi,

$$\begin{aligned}\widehat{S}(T_{Y/X}) &= \cdots + \frac{(n_{.i}+n_{.j})}{n}\sum_{k=1}^{K}\frac{n_{ki}+n_{kj}}{n_{.i}+n_{.j}}\log(\frac{n_{ki}+n_{kj}}{n_{.i}+n_{.j}}) + \cdots \\ &= \widehat{S}(T''_{Y/X})\end{aligned}$$

Donc, nous trouvons bien

$$\Delta\widehat{S}(T_{Y/X}) = \Delta\widehat{S}(T''_{Y/X})$$

Conclusion:

La distance de Mantaras et le gain ratio ne remplissent pas la propriété 5, nous retrouvons analytiquement les résultats expérimentaux de [White et Liu, 1994] : ces mesures ne pénalisent pas les attributs multivalués lors de la sélection des variables pour l'éclatement d'un noeud. Certes la propriété 5 décrit une situation limite que l'on rencontre très peu dans la pratique (égalité des distributions empiriques des classes entre deux colonnes du tableau $T_{Y/X}$), elle caractérise néanmoins les incapacités théoriques de la mesure.

3.9 Etudes empiriques : quelles sont les meilleures mesures ?

Face à la multiplicité des mesures de qualité de partitions, il y a dans la littérature de nombreuses études empiriques autour de leur efficience dans la construction des graphes d'induction. Contrairement à ce qu'on pourrait penser, sans aller quand même jusqu'aux conclusions de [Mingers, 1989b], **il n'y a pas de critères meilleurs que les autres pour peu qu'ils répondent aux propriétés de minimalité et de maximalité, du moins en terme de performances en classement du modèle**. Cette conclusion déjà ancienne [Baker et Jain, 1976], a été relayée par [Breiman *et al.*, 1984] qui conjecturaient que seul le choix de la complexité du modèle avait une importance véritable en classement. Ce point de vue a été confirmé par la plupart des expérimentations à grande échelle [Lerman et Costa, 1996] [Buntine et Niblett, 1992] [Utgoff et Clouse, 1996], même si certaines études ponctuelles ont montré que dans certaines situations certaines mesures pouvaient mieux se comporter [Fayyad et Irani, 1992].

Est-ce que pour autant le sujet est clos, et que la recherche à ce sujet est vouée à l'échec?

Non, définitivement non. Car si en terme de précision, les différents indicateurs sont à peu près au même niveau, il reste que la définition de la mesure permet de guider l'induction et de donner au graphe des propriétés désirées selon le problème à résoudre ou l'interprétation que l'on veut lui donner. Par exemple, certains auteurs se sont aperçus que l'indice de Gini favorisait les sommets enfants de taille et de pureté assez équilibrées [Taylor et Silverman, 1993], et que le gain ratio en revanche, notamment en discrétisation binaire, préférait les découpages mettant en exergue des petits sommets où la pureté est forte, l'arbre de décision prenant alors la forme d'un

"râteau"[Kohavi et Sahami, 1996]. D'autres auteurs encore se sont penchés sur la spécification de la mesure dans des circonstances particulières d'apprentissage : données bruitées, attributs manquants, et lorsque le nombre de classes est élevé [Kira et Rendell, 1992] [Kononenko, 1994].

Au final, un des principaux enjeux des mesures est la capacité à explorer au mieux les solutions les plus générales pour construire les graphes de la plus petite taille possible. En ce sens, il faut que le critère soit suffisamment stable pour ne pas être perturbé par les faibles fluctuations d'échantillonnage, ce qui permettrait d'apporter une solution au problème de la forte variance des graphes [Wehenkel, 1997]. En tous les cas, de la plupart des analyses empiriques, nous constatons que finalement les mesures classiques telles que l'indice de Gini, le Gain ratio, la distance de Kolmogorov-Smirnov, le Gain d'entropie ou encore la mesure du χ^2 figurent parmi les meilleurs compromis, et que les mesures "tarabiscotées", empreintes de différentes normalisations à partir du gain classique se révèlent finalement très instables [Ho et Nguyen, 1996].

3.10 Conclusion

Prétendre à l'exhaustivité était particulièrement difficile dans ce chapitre. Il existe encore une multitude de mesures qui n'entrent pas dans la classification ci-dessus. La plupart néanmoins sont dérivées des entropies généralisées de [Daroczy, 1970], et n'apportent que des modifications mineures qui finalement s'avèrent peu décisives dans la pratique.

Si en performances pures, les mesures se valent, les graphes induits diffèrent par leur taille. Nous pensons que c'est un résultat très important, en spécifiant les propriétés désirées, nous pouvons a priori avoir une idée sur les caractéristiques du classifieur induit. A l'inverse, dans des situations particulières, selon nos exigences en terme d'exploration et de configuration de l'arbre, nous pourrons choisir la mesure la plus appropriée. Quoiqu'il en soit, ces nombreuses études ont certainement contribué à mieux comprendre le comportement et le rôle de cette composante dans l'induction, notamment dans l'exploration des solutions candidates.

Enfin, nous constatons que la caractérisation des règles que nous avons adopté dans la section 3.3.2 se révèlent efficaces dans la pratique pour distinguer une catégorie de mesures destinés à éviter la fragmentation des données. Nous pouvons dès lors mettre à profit ces résultats pour définir non plus des mesures dévaluation d'une segmentation locale mais plutôt des mesures d'évaluation de partitions globales. Les différentes configurations de graphes sont ainsi interprétés comme des super-attributs candidats, les propriétés de fusion et de sensibilité à la taille de l'effectif garantissent l'absence de biais en faveur des classifieurs comportant beaucoup de feuilles comme nous le verrons plus loin dans cette thèse.

Chapitre 4

Détermination de la taille optimale du graphe d'induction

4.1 Introduction

La détermination de la taille optimale des graphes est un élément crucial de la construction du classifieur. L'optimalité est complètement subjective et obéit en réalité à deux objectifs qui peuvent être à la fois antagonistes et concomitants.

Le premier objectif est de construire le classifieur le plus simple pour des raisons de compréhensibilité [Cherkauer et Shavlik, 1996] et de rapidité. En effet, un des principaux avantages des graphes par rapport à d'autres systèmes d'apprentissage est de produire un classifieur qui nous permet d'expliquer les relations de causalité dans le domaine étudié. Il nous faut toujours garder à l'esprit qu'un modèle n'est vraiment applicable que s'il a convaincu ses utilisateurs. En fournissant les tenants et aboutissants de leur prise de décision, les graphes offrent une opportunité supplémentaire de validation surtout lorsqu'ils sont appliqués dans des domaines critiques comme la médecine.

Toujours en faveur de la simplicité, on aimerait que le classifieur soit le plus rapide en généralisation. Dans cette optique, [Tan et Schlimmer, 1990] ont défini la "complexité dynamique", qui correspond au nombre de questions posées en moyenne pas le classifieur pour déterminer la classe d'un individu. Il est clair que dans la plupart des cas, pour des motifs de coûts et de rapidité de traitement, on a intérêt à produire le modèle le plus simple. Par exemple, aux urgences d'un hôpital, plus le nombre de questions à poser sera faible, plus vite les médecins pourront intervenir. Pour ces raisons, si deux classifieurs présentent les mêmes performances en généralisation, nous préférerons toujours le plus simple.

Le second objectif est justement en relation avec la notion de performances en classement. Pendant longtemps, on a reproché aux arbres leur propension à construire des classifieurs com-

plexes, collant éxagérement aux données et dont le taux de succès s'effondrait en généralisation. C'est le phénomène de sur-apprentissage[13]. [Breiman *et al.*, 1984] ont été parmi les premiers à arguer que la qualité de l'arbre dépendait plus de sa taille que de la mesure de segmentation utilisée dans le choix des attributs discriminants lors du partitionnement des individus. Ils ont entres autres proposé une méthode pour y remédier, l'élagage, qui a fait énormément d'émules ces dernières années.

Dans ce chapitre, nous nous intéresserons plus particulièrement à cette deuxième visée. Nous exposerons dans un premier temps la problématique de la recherche de la taille adéquate du graphe. Dans un second temps, nous présentons les principales pistes justifiant le choix de la complexité : d'abord la préférence à la simplicité fondée à la fois sur le principe du rasoir d'Occam et l'appréhension de données bruitées, qui fût par la suite battu en brèche par des chercheurs qui affirmèrent qu'un tel choix est avant tout dépendant du domaine et repose sur des connaissances a priori. A la lumière de ces développements, nous présenterons les principales heuristiques développées depuis le milieu des années 80.

4.2 Problématique de la taille optimale

A la faveur d'expérimentations à grande échelle, [Breiman *et al.*, 1984] ont observé qu'à mesure que la taille de l'arbre augmentait en induction, le taux d'erreur en resubstitution (mesuré sur l'échantillon d'apprentissage) diminuait constamment, fortement d'abord lorsque l'on est proche de la racine, suivant une pente plus douce ensuite. En revanche, le taux d'erreur en généralisation (mesuré sur un échantillon test à part), après une diminution jusqu'à un point optimal, remontait par la suite (figure 4.1). Par analogie avec la régression où l'on constate qu'à partir d'un certain seuil, l'adjonction de variables supplémentaires augmente constamment le coefficient de corrélation empirique au détriment du vrai coefficient de corrélation que l'on veut maximiser, les auteurs soumettent une analyse en terme de décomposition du biais et de la variance statistique dont nous reprenons ici les principaux traits.

Soit un problème à deux classes avec les distributions a priori $[\pi(1), \pi(2)]$ et conditionnelles $f_j(x)$ $(j = 1, 2)$[14]. Le taux d'erreur bayésien optimal est

$$R^* = 1 - \int \max_j(\pi(j).f_j(x)).dx$$

L'espace de représentation $\aleph$ est subdivisé en L rectangles $S_1, \ldots, S_L$ correspondant aux sommets terminaux de l'arbre. A chaque feuille l est assignée une classe conclusion c_l.

Par définition, le "vrai" taux d'erreur associée à la partition $S_1, \ldots, S_L$ s'écrit

$$R^*(L) \quad = \quad \sum_l P(X \in S_l, Y \neq c_l)$$

13. overfitting

14. Il s'agit ici de la fonction de densité de la probabilité $P(X = x/Y = j)$

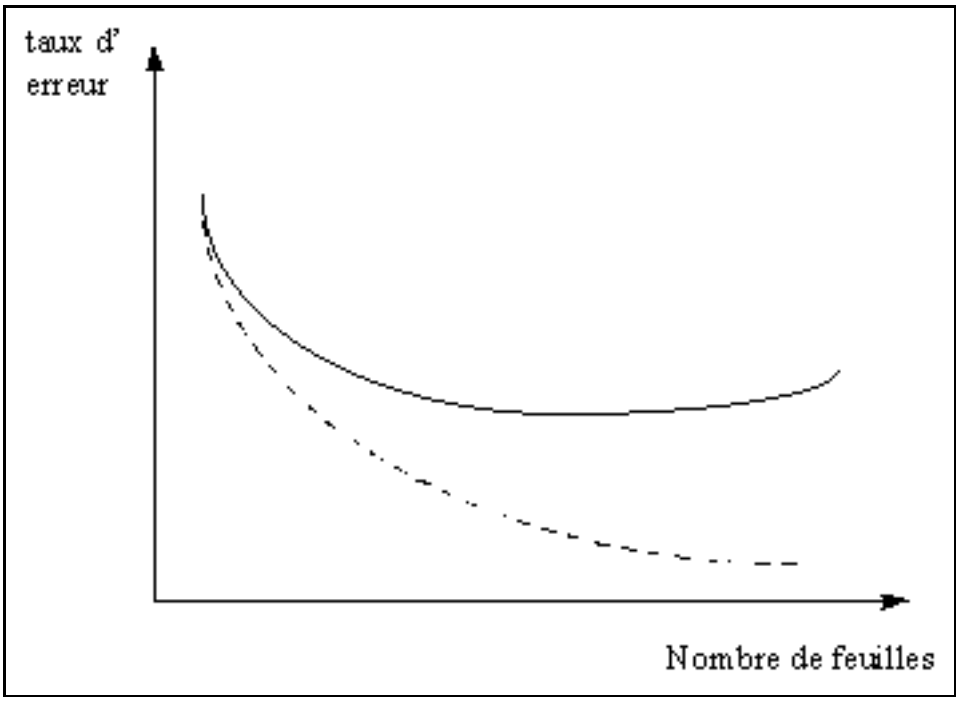

FIG. 4.1 – *En pointillés, l'évolution du taux d'erreur en resubstitution; en continu, le taux d'erreur sur l'échantillon test*

$$= \quad 1 - \sum_{l} P(X \in S_l, Y = c_l) \tag{4.1}$$

Soit $\tau(condition)$ une fonction indicatrice prenant la valeur 1 si *condition* est vrai, 0 sinon. L'équation 4.1 s'écrit

$$R^*(L) = 1 - \sum_{l,j} \tau(c_l = j).P(X \in S_l, Y = j) \tag{4.2}$$

En adoptant la règle d'affectation bayesienne minimisant le coût de mauvais classement espéré

$$y_l^* = \begin{cases} 1, \ si \ \ P(X \in S_l, Y = 1) > P(X \in S_l, Y = 1) \\ 2, \ sinon \end{cases}$$

on peut ainsi écrire l'équation 4.2 de la manière suivante

$$\begin{aligned} R^*(L) \quad = \quad & (1 - \sum_{l} \max_{j} P(X \in S_l, Y = j)) \\ & + \sum_{l} \tau(y_{l \neq} y_l^*) \ |P(X \in S_l, Y = 1) - P(X \in S_l, Y = 2)| \end{aligned} \tag{4.3}$$

[Breiman *et al.*, 1984] considèrent les deux termes de l'équation 4.3 comme un biais et une variance[15] dont voici une interprétation :

– le biais représente l'erreur de l'hypothèse moyenne de l'arbre, elle correspond en quelque

15. nous expliciterons plus en détail la décomposition biais-variance dans le chapitre sur les graphes multiples. Les nouvelles formulations diffèrent sensiblement de ce qui est présenté ici, l'interprétation en revanche reste identique.

sorte à l'écart entre l'arbre construit où les affectations aux feuilles sont optimales (T_{biais}) avec le meilleur arbre que l'on puisse construire sur ces données;

– la variance, elle, correspond alors à l'écart moyen entre l'erreur de l'arbre actuel avec celle de T_{biais}.

[Breiman *et al.*, 1984] montrent alors qu'avec l'augmentation de la complexité de l'arbre, le biais diminue fortement pour des faibles valeurs de L, plus lentement par la suite, alors que dans le même temps, la variance augmente, parce que l'affectation d'une classe à une feuille dépend de plus en plus d'une fraction plus petite de l'échantillon. La taille optimale de l'arbre sera alors celle où l'addition de ces deux termes est minimum.

Notons que s'il est d'usage d'exhiber la courbe en U montrant l'augmentation de l'erreur totale après un point minimum, rien ne garantit qu'il en est toujours ainsi dans la pratique [Schaffer, 1993a]. Il se peut très bien alors que le biais diminue alors que la variance reste faible, en fait tout dépend de la nature des données étudiées (niveau de bruit, ...).

Sous d'autres formulations, ce problème est bien connu en apprentissage. [Shahshahani et Landgrebe, 1994] ont noté en reconnaissance de formes que lorsque le ratio entre la taille d'échantillon et la dimension de représentation est trop faible, la fonction discriminante est mal estimée. Cela serait dû au fait que l'estimation des densités conditionnelles étant produite avec l'échantillon d'apprentissage, la diminution de l'erreur bayesienne estimée est contrebalancée par le biais d'erreur de classement. Lorsque l'accroissement de ce dernier est plus fort que la diminution de l'erreur bayesienne, l'utilisation d'une variable supplémentaire dégrade les performances du classifieur [Hughes, 1968].

[Vapnik, 1982], enfin, analyse cette évolution sous l'angle de la "minimisation du risque structurel". Selon lui, parmi les différents modèles candidats de complexité croissante, il en existe un qui minimise le risque global, correspondant à la sommation du risque empirique et du risque espéré lorsque que l'on classe respectivement les individus de l'échantillon d'apprentissage et les individus de l'échantillon test. Si le classifieur est trop simple, on s'expose à "l'erreur approximée" où l'on étiquette mal les observations de l'apprentissage; en revanche si le classifieur est trop complexe, "l'erreur estimée" peut être éxagérée i.e le classifieur collant trop au données, reconnaît mal les nouveaux individus.

4.3 Préférences pour les graphes de petite taille

4.3.1 Le rasoir d'Occam

Le principe du rasoir d'Occam tient en une phrase: "non sunt multiplicanda entia praeter necessitatem" que l'on peut traduire littéralement par "les entités ne sont pas à multiplier outre mesure" [Blumer *et al.*, 1987]. C'est un postulat extrêmement séduisant que l'on a souvent déchiffré

sous la forme d'une préférence pour la simplicité [Fisher et Schlimmer, 1988] avec comme l'idée sous-jacente qu'un classifieur de complexité moindre consistant sur la même base de données sera plus robuste en généralisation [Nunez, 1988] [Tan et Schlimmer, 1990]. [Fayyad et Irani, 1990] en ont d'ailleurs proposé une démonstration formelle sur des arbres binaires.

Il existe de nombreux travaux autour de cette question, un des plus intéressants est sans conteste celui de [Blumer *et al.*, 1987] où les auteurs montrent que pour un modèle simple, il est possible de garantir les performances en généralisation sur la base de l'échantillon d'apprentissage. Leur démonstration repose sur le modèle mathématique PAC (Probability Approximately Correct) de [Valiant, 1984]. Il en résulte une préférence en faveur des modèles simples, d'autant plus qu'étant plus rares dans l'espace des hypothèses, il y a très peu de chances qu'ils soient consistants par hasard sur les données [Russel et Norvig, 1995].

Dans le même ordre d'idée, [Auer *et al.*, 1995] ont proposé un classifieur sous la forme d'un arbre de décision à deux niveaux, dont la VC-dimension[16] est finie. Ils ont montré qu'il était possible dès lors de borner l'erreur sur l'échantillon test. Ces résultats deviennent caduques avec l'augmentation de la taille de l'arbre, la VC-dimension tend vers l'infini, et il n'y a plus aucune garantie sur la borne de l'erreur.

De manière plus générale, le rasoir d'Occam justifie également le principe de l'arbitrage entre généralité et précision que l'on retrouve dans les approches de description minimale et assimilées en apprentissage [Rissanen, 1978] [Freeman, 1985].

4.3.2 Apprentissage sur données bruitées - Traitement du sur-apprentissage

La plupart des méthodes en apprentissage automatique ont été conçues pour des domaines d'étude dépourvus de bruit. De fait, la problématique était essentiellement de trouver des classifieurs qui soient consistants sur les données d'apprentissage i.e tous les individus sont classés correctement.

Sur des bases réelles, une telle attitude n'est plus soutenable. La description du concept à trouver n'est plus parfaite car les données présentent des incertitudes [Kodratoff et Manago, 1987] : soit sous la forme de bruit (erreur sur la classe, erreur sur un ou plusieurs attributs descriptifs, soit un mélange des deux), soit parce que certaines variables déterminantes manquent. De fait, l'apprentissage parfait est impossible, et même indésirable dans certains cas, il faut savoir distinguer entre les formes à reconnaître et les phénomènes qui ne relèvent que du bruit.

La principale répercussion des données bruitées sur la construction des graphes d'induction est le sur-apprentissage [Quinlan, 1986a][Watkins, 1987], les graphes sont de taille éxagérée car on introduit des variables supplémentaires qui semblent pertinentes au vu de l'échantillon d'ap-

16. la VC-dimension, issu de travaux de [Vapnik, 1982], traduit la richesse d'expression d'un modèle. Elle correspond au nombre d'observations différentes qu'il peut entièrement décrire. Par exemple, pour un arbre booléen à deux niveau avec deux attributs prédictifs, la VC-dimension est 4.

prentissage alors qu'en réalité elle ne le sont pas dans la population mère. Dans le cas particulier des arbres, il est communément admis que le bruit est le plus pernicieux dans les parties basses de l'arbre, lorsque les individus associés aux sommets sont faibles, diminuant fortement la fiabilité de l'action entreprise : que ce soit la segmentation sur un noeud ou l'assignation d'une classe à une feuille. De nombreuses méthodes de limitation de la taille des graphes ont ainsi vu le jour pour traiter convenablement le bruit, ces techniques reposent sur des justifications plus ou moins discutables, la démonstration de leur efficacité étant établie avant tout sur des expérimentations sur données artificiellement bruitées et réelles [Mingers, 1989a].

En tous les cas, encore une fois, il est généralement admis que pour traiter le bruit, on doit limiter la taille du graphe.

4.4 Le traitement du sur-apprentissage résultant de choix délibérés

4.4.1 Refus de l'universalité de la préférence à la simplicité

La diffusion des bases de données benchmark a amené plusieurs chercheurs, quasiment simultanément, à évaluer empiriquement les principes ci-dessus, notamment les arbres les plus simples composés d'un seul niveau [Iba et Langley, 1992][Holte, 1993]. Remarquant que souvent dans les études expérimentales, la réduction drastique de la taille des arbres amenait au pire une faible réduction du taux de succès en validation, ils ont mené une étude exhaustive du comportement d'un arbre à un niveau face à des algorithmes rodés comme C4.5 [Quinlan, 1993a]. Il est apparu alors que le gain du passage en performance à un modèle plus complexe était finalement minime au regard de la simplicité du modèle sur 16 bases de données de l'UCI Irvine [Murphy et Aha, 1995]. Leur conclusion était que sur des bases de données réelles, la préférence devrait toujours être donnée à des modèles très simples.

Des affirmations aussi péremptoires ne pouvaient pas rester longtemps sans réponse, et le papier de [Holte, 1993] notamment provoqua une levée de boucliers qui eût pour mérite d'éclaircir sous un autre jour le problème de la taille optimale dans les graphes d'induction.

[Elomaa, 1994], tout d'abord, a critiqué point par point le papier de [Holte, 1993], il mit surtout en évidence le peu de représentativité des données utilisées par ce dernier qui prétendait définir le concept de "données naturelles" à partir de l'absence de préparation pour l'apprentissage. Toutes les bases sont le fruit de manipulations plus ou moins élaborées, dans les ECD par exemple, elles sont optimisées pour le traitement informatique (normalisations, élimination de la redondance...), et se présentent souvent sous une forme tabulaire propice à l'apprentissage [Agrawal *et al.*, 1992]. Doit-on pour autant les qualifier de "naturelles" ? En fait, la principale conclusion de ce papier a été de montrer que les bases benchmark de la communauté de l'intelli-

gence artificielle étaient trop souvent "simples", en tous les cas pour la plupart correspondant à des arbres de petite taille.

Plus en rapport avec nos préoccupations ont été les séries de papiers produits par [Schaffer, 1993a]. A partir d'études sur données artificielles, il a balayé tous les arguments en faveur de la simplicité des graphes en exhibant des contre exemples à toutes les affirmations précédentes. Ces résultats furent relayés par ceux de [Murphy et Pazzani, 1994] qui montrèrent, également sur données artificielles, qu'en testant tous les classifieurs consistants sur les données, il apparaissait clairement que les arbres les plus simples n'étaient pas les plus performants en généralisation, réfutant ainsi les précédentes interprétations du rasoir d'Occam. Plus fort encore, ils montrèrent également qu'en bruitant les données, la préférence à la simplicité n'est pas la meilleure attitude, tout dépend de la nature du bruit.

Prenons un exemple pour illustrer leur propos. Nous avons à étudier le concept suivant

$$f = x_1 \, XOR \, x_2 \tag{4.4}$$

L'arbre minimum pour décrire l'équation 4.4 comporte 4 feuilles, si nous disposons de quatre individus dont la description couvre parfaitement le concept, nous devrions pouvoir le reconstituer :

	X_1	X_2	Y
1	$vrai$	$vrai$	$faux$
2	$vrai$	$faux$	$vrai$
3	$faux$	$vrai$	$vrai$
4	$faux$	$faux$	$faux$

En vérité, nous connaissons très peu d'algorithmes qui, avec leurs paramètres standards, accepteraient de construire un arbre booléen comportant 4 feuilles avec quatre individus. La plupart du temps, la partition sera refusée car portée par trop peu d'individus, les conclusions seront assimilées à du bruit.

La thèse de ces auteurs repose sur le postulat d'impossibilité de spécifier une préférence pour la simplicité à partir de données, ils pensent que ce choix est dicté avant tout par des considérations a priori. Leur principale contribution est d'avoir déterminé, hélas souvent empiriquement, les situations dans lesquelles il était profitable de favoriser soit les graphes de petite taille, soit les graphes de grande taille.

4.4.2 Caractérisation des préférences en fonction des connaissances a priori

Les travaux de [Buntine, 1992] sur l'application de la théorie bayesienne en induction constituent les prémices des études que nous présenterons ci-après : intuitivement, l'auteur opte pour une distribution a priori en fonction de la nature des données qu'il a à traiter. Avec les papiers

de [Schaffer, 1991] [Schaffer, 1992] [Murphy, 1995], on a pu mieux cerner les caractéristiques que l'on devait considérer pour choisir le meilleur biais. Les résultats sont pour la plupart empiriques, elles ont pour mérite de remettre en question le paradigme de la préférence pour la simplicité dans la construction des graphes d'induction.

Les préférences en fonction de la relation Complexité du concept et Quantité de données disponibles

Dans sa thèse concernant l'induction de données sur de très grosses bases de données, [Catlett, 1991a] avait constaté qu'il était préférable, plutôt que de procéder à un échantillonnage, d'apprendre sur la totalité du fichier en limitant explicitement la taille de l'arbre. [Oates et Jensen, 1997] également a remarqué que la taille des arbres de décision croissait avec l'effectif de l'échantillon d'apprentissage et propose des techniques de filtrages par élimination des observations non informatives [John, 1995] [Brodley et Friedl, 1996].

A la lumière de la section précédente, ces constats prennent une nouvelle dimension. En effet, [Schaffer, 1992] et [Fisher, 1992] ont montré qu'il était bénéfique de privilégier la simplicité lorsque le rapport entre la taille de l'échantillon et la complexité du concept à étudier est relativement fort, cela inclut la situation où l'on essaie d'apprendre sur des données sur lesquelles on suspecte un grand nombre d'attributs non-pertinents [Murphy, 1995], ce qui est le cas généralement lorsque l'on explore pour la première fois des bases de données fourre-tout, et que l'on ne connaît pas véritablement quelles sont les causalités attendues. Bien entendu, dans le cas contraire, lorsque les données sont peu nombreuses, on a intérêt à ne pas brider la construction de l'arbre, et privilégier une plus grande taille surtout lorsque l'on s'attend à ce que le concept à étudier soit difficile.

Nous devons bien faire attention à la notion de complexité qui est avancée ici. Elle doit être comprise relativement à la puissance de représentation du modèle. Prenons un exemple simple pour mieux comprendre cette distinction. L'arbre minimum pour décrire la fonction booléenne $f = x_1x_2 + x_3x_4$ contient 7 feuilles. Si nous ne disposons que d'une dizaine d'individus, il est évident que nous devons favoriser les arbres de grande taille pour espérer approcher le bon concept. En revanche, si nous passons à un système de représentation plus puissant, les graphes comme nous le verrons dans un chapitre prochain, le classifieur optimal ne contient que 3 feuilles, il est préférable dès lors d'opter pour la simplicité.

[Schaffer, 1992] a étendu son étude sur des bases réelles de l'UCI Irvine [Murphy et Pazzani, 1994]. En comparant les performances de CART [Breiman *et al.*, 1984], avec et sans élagage, il montre des résultats édifiants où il apparaît qu'à mesure que l'on diminue la taille de l'échantillon d'apprentissage, les performances en généralisation du second algorithme rattrapent puis dépassent ceux du premier.

Bien que l'auteur n'en fasse pas mention dans ses travaux, nous avons noté que dans le même temps l'écart relatif entre la taille de l'arbre après et avant élagage se restreint, ce qui à notre sens indique que la préférence à la simplicité n'est pas aussi forte que l'on puisse penser dans CART et que c'est une procédure relativement robuste compte tenu des conditions extrêmes auxquelles elle a été soumise dans cette expérimentation.

Les préférences en fonction de la quantité et de la nature du bruit

Le second volet sur lequel les auteurs sus-cités ont travaillé est l'influence du bruit. Il est communément admis qu'il induit le sur-apprentissage qui se manifeste par des tailles éxagérées des graphes.

En fait, on doit distinguer deux types de bruits, leur effet est différent sur la préférence à la simplicité [Schaffer, 1991] [Schaffer, 1993a] :

- le danger du sur-apprentissage est manifeste lorsque le bruit porte sur les attributs; à la limite lorsqu'ils sont non-pertinents, on peut s'attendre à un partitionnement en pure perte de l'espace de représentation. Dans ce cas, il est conseillé de choisir les modèles les plus simples.

- en revanche lorsque le bruit affecte la variable à prédire, les avis sont partagés. [Schaffer, 1993a] pense que la préférence à la complexité est payante, ceci d'autant plus que le niveau du bruit augmente. En effet on constate dans ce cas que les individus sont bien dirigés vers les bons sommets du graphe mais que leur étiquetage est perturbé par le bruit, et l'on obtiendra des mauvais taux de bon classement en resubstitution. De tels sommets seront immanquablement supprimés à tort par les procédures d'élagage fondées sur des estimations plus ou moins sophistiquées du taux d'erreur. Mais on se rend compte que si l'on pousse le raisonnement à l'extrême avec un bruitage maximal, il est évident que l'expansion de l'arbre propagera le même niveau de bruit dans les feuilles, quelles que soient leurs tailles. De fait, les classes resteront indiscernables, il est préferable de refuser la construction du graphe.

Ces derniers résultats ne remettent pas en cause la préférence à la simplicité, ils contestent plutôt leur universalité [Buntine, 1990] [Weiss et Indurkhya, 1994]. Même si un paradigme donne de "bons résultats" sur la plupart des études empiriques, il apparaît ici qu'il est encore plus bénéfique de s'intéresser à ses conditions d'applications. Il reste néanmoins que la préférence à la simplicité est une excellente approche des données lorsque l'on dispose de peu d'informations, elle est d'autant plus intéressante que pour des motifs de compréhensibilité et de maniabilité exposés en introduction, il est toujours préférable de choisir le plus simple parmi les modèles qui présentent les même performances **en généralisation.**

Dans ce qui suit, nous allons présenter les différentes heuristiques de détection de la bonne taille des arbres en induction, nous réservons au chapitre suivant les heuristiques applicables à la généralisation aux graphes.

4.5 Critères d'arrêt de l'expansion de l'arbre

C'est la stratégie la plus naturelle, elle exploite le fait que la partition soit récursive, chaque noeud peut être évalué indépendamment des autres. Il y a deux manières de le faire : soit en considérant ses qualités intrinsèques (pureté, effectifs...), soit en considérant ses éventuels sommets enfants (significativité de la segmentation, contrainte d'admissibilité...).

A l'actif du critère d'arrêt se trouve essentiellement la rapidité de traitement, le nombre de tests à effectuer diminue de manière drastique, les variables peu pertinentes sont d'office éliminées. A son passif, certains auteurs [Breiman *et al.*, 1984] lui reprochent souvent la difficulté à fixer les bonnes valeurs des paramètres lors de l'induction.

4.5.1 Identification d'une feuille

Le premier critère le plus évident est la pureté du noeud, lorsqu'il ne contient que des représentants d'une classe, il est évident que le partitionnement qui suit ne peut être informatif. Certains auteurs [Wehenkel, 1990] [Cestnik *et al.*, 1987a] l'ont étendu à la définition d'une valeur minimale $H_{\min}$, dès que l'entropie calculée sur le sommet $\widehat{S}(T_Y)$ est en-dessous de ce seuil, on arrête la segmentation du noeud. A priori, la valeur $H_{\min}$ est fixée par l'utilisateur, elle est indépendante de la feuille que l'on étudie, nous verrons dans la section suivante que l'on peut trouver une valeur de $H_{\min}$ automatiquement.

Le second critère pour identifier une feuille est sa taille. Si celle-ci est inférieure à une valeur $n_{\min}$, on refuse de subdiviser le noeud car l'on pense que la partition qui en résultera sera trop fine et ne reflétera en aucune manière une quelconque structure de causalité dans les données. Dans CART et SIPINA [Breiman *et al.*, 1984] [Zighed *et al.*, 1992], cette valeur est fixée à 5 par défaut, dans C4.5 elle est à 2 [Quinlan, 1993a]. Dans certains cas on étend ce critère à l'admissibilité des partitions enfants : si un ou plusieurs sommets enfants candidats présentent des feuilles de taille inférieure à la limite $n_{\min}$ on refuse la segmentation même si par ailleurs les autres sommets sont très informatifs (figure 4.2). Le choix de la valeur de coupure absolue $n_{\min}$ pose souvent problème, en réalité on devrait la fixer selon le domaine étudié et surtout la taille de l'échantillon initial, il est évident qu'une taille minimale de 5 individus est concevable si l'on dispose de 100 individus, en revanche si nous n'en disposons que de 10, $n_{\min} = 5$ est franchement mal défini. Dans cette optique, [Cestnik *et al.*, 1987a] préfèrent fixer un rapport limite entre la taille de la feuille et l'échantillon initial.

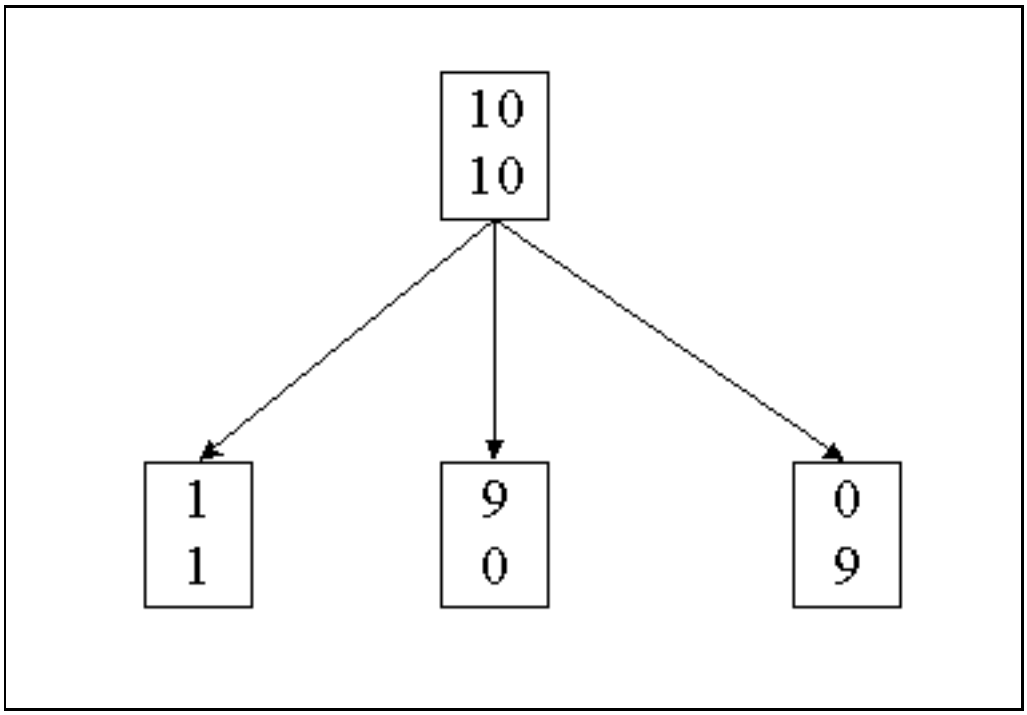

FIG. 4.2 – *Avec une contrainte d'admissibilité fixée à 5 individus, la partition sera refusée à cause du sommet à gauche*

Dans la pratique, l'obligation de taille minimale s'avère trop restrictive et peu robuste [Murthy, 1995], il est plus judicieux de laisser se développer l'arbre et d'utiliser des heuristiques de regroupement pour éviter la constitution des sommets à trop faibles effectifs, ce qui de plus résout la question des découpages où l'une des branches ne contient aucun individu tout simplement parce la modalité de l'attribut prédictif est rare.

4.5.2 Critère d'arrêt sur la segmentation - Pré-élagage

La formulation originelle

La question est de savoir si la segmentation que l'on est en train d'effectuer sur le noeud en cours apporte significativement de l'information. Dans le cas du gain d'entropie de Shannon, nous savons que $\Delta S(T_{Y/X}) \geq 0$ quel que soit le découpage, si la valeur mesurée $\Delta\widehat{S}(T_{Y/X})$ s'avère "trop faible" sur un noeud, nous considérons que le partitionnement n'est pas acceptable.

Comment appréhender la notion "trop faible" ? Il pourrait paraître intuitif de fixer la valeur seuil ΔS_{seuil} à 0 car c'est le cas théorique où les distributions de probabilités sur les sommets enfants sont toutes identiques. Hélas, ce serait méconnaître l'influence des fluctuations d'échantillonnage et du bruit, un seul individu atypique (ne correspondant pas à un concept déterministe) amène inévitablement $\Delta\widehat{S}(T_{Y/X})$ à des valeurs non-nulles. Nous verrons plus loin que l'on peut

quantifier la valeur moyenne de $\Delta\widehat{S}(T_{Y/X})$ en l'absence totale de liaison entre les attributs $X(.)$ et $Y(.)$.

La première solution avancée par certains auteurs est de fixer une valeur seuil ΔS_{seuil} suffisamment grande pour discerner un éventuel phénomène de causalité du simple bruit. Sa détermination répond alors à des considérations empiriques, et repose essentiellement sur l'expérience de l'expert.

En fait, nous pouvons adopter une démarche autrement plus rigoureuse dans le choix de ce seuil en adoptant la formulation d'un test statistique. L'hypothèse nulle correspond à la situation d'indépendance entre $X(.)$ et $Y(.)$,

$$\begin{aligned} H_0 &: X(.) \textit{ et } Y(.) \textit{ sont indépendants} \\ H_1 &: X(.) \textit{ et } Y(.) \textit{ sont liés} \end{aligned}$$

Plusieurs statistiques peuvent être utilisées, le test le plus connu est sans aucun doute celui du χ^2 qui est introduit comme règle d'arrêt dans ID3 [Quinlan, 1986b] et ChAID [Kass, 1980]. Mais il est plus cohérent à notre sens de proposer une règle d'arrêt basée sur la mesure de qualité de partition employée lors de l'expansion de l'arbre. En ce qui concerne le gain d'entropie de Shannon, nous pouvons en dériver la statistique $G(T_{Y/X})$ [Mingers, 1989b] [White et Liu, 1994] telle que

$$G(T_{Y/X}) = 2 \times n \times \ln(2) \times \Delta\widehat{S}(T_{Y/X}) \tag{4.5}$$

Sous l'hypothèse nulle, elle suit une loi de χ^2 de $(K-1)(L-1)$ degrés de liberté

L est le nombre de modalités de la variable $X(.)$, dès lors la valeur seuil de rejet de l'hypothèse nulle pour un risque de première espèce α (la probabilité de rejeter l'hypothèse nulle alors qu'elle est vraie) est tout simplement la valeur lue dans la table du χ^2 au point de pourcentage $1-\alpha$.

$$G_{seuil} = \chi^2_{1-\alpha}(K-1)(L-1)$$

Nous pouvons dès lors déterminer ΔS_{seuil}

$$\Delta S_{seuil} = \frac{\chi^2_{1-\alpha}(K-1)(L-1)}{2 \times n \times \ln(2)}$$

Bien entendu, à l'instar de tous les tests statistiques, nous pouvons réécrire la région critique comme suit

$$P(\chi^2 > G(T_{Y/X})) = \alpha_{obs}$$

$$\textit{Décider } H_1 \textit{ si } \alpha_{obs} < \alpha \tag{4.6}$$

Ce résultat amène plusieurs commentaires :

- conformément à notre remarque précédente pour la détection d'une feuille en fixant une valeur limite $H_{\min}$, puisque $\widehat{S}(T_Y) > \Delta\widehat{S}(T_{Y/X})$, on peut se dispenser de tenter la segmentation si

$$\widehat{S}(T_Y) \leq \Delta S_{seuil}$$

- comme le fait remarquer [Wehenkel, 1990], nous constatons qu'en moyenne sous l'hypothèse nulle, le gain d'entropie est supérieur à 0, ce qui confirme la nécessité d'une valeur seuil positive

$$\begin{aligned} E(\Delta\widehat{S}(T_{Y/X})) &= \frac{(K-1)(L-1)}{2 \times n \times \ln(2)} \\ &\geq 0 \end{aligned}$$

- la préférence à la simplicité est ainsi exprimée par le choix convenable de α :
  - lorsqu'il est très faible, la valeur seuil ΔS_{seuil} devient extrêmement élevée et l'arbre produit est en général de petite taille;
  - dans le cas contraire, le seuil est faible et l'on produit des grands arbres.

- il n'y a pas de règle générale pour fixer une valeur standard de α, certes l'usage en statistique nous incite à fixer des valeurs couramment usitées (0.1, 0.05, 0.01, ...) mais en fait tout dépend du domaine analysé.

- notons qu'implicitement, le choix de la règle d'arrêt fondé sur la pertinence de la segmentation traduit une préférence à la simplicité. En effet, rien n'annonce le long d'une branche de l'arbre que l'évolution de la probabilité critique associé au test sera monotonement croissante. Si nous disposons de deux solutions possibles, il est clair que la seconde ne sera jamais explorée puisque l'expansion de l'arbre sera stoppée avant. En prenant l'exemple de la construction d'un arbre de décision sur le fichier des ondes de [Breiman *et al.*, 1984], nous le constatons empiriquement (figure 4.3).

Correction de Bonferroni pour les tests de significativité du lien

La spécification du test ci-dessus n'est pas à proprement parler fausse, mais nous avons omis de signaler une chose extrêmement importante : la variable $X(.)$ a été choisie parmi les $X_1(), \ldots, X_p()$ variables tel que la mesure d'évaluation du partitionnement $G(T_{Y/X_i})$ est maximum.

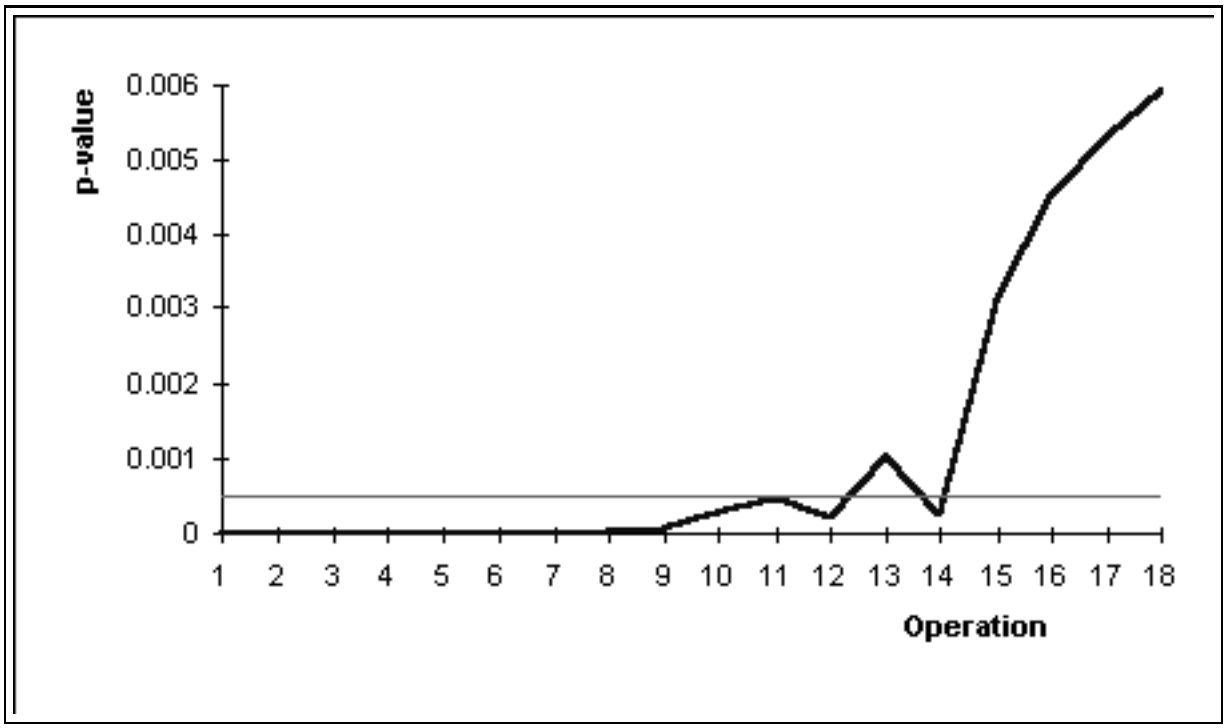

FIG. 4.3 – *Evolution du risque critique du test d'arrêt selon le numéro de l'opération de segmentation : si le seuil est fixé à* α=0.0005*, nous observons deux solutions possibles*

La spécification du test ci-dessus change, l'hypothèse nulle n'est plus

$$H_0 : X(.) \; et \; Y(.) \; sont \; indépendants$$

mais

$$H_0{:}X(.) \; est \; tel \; que \; G(T_{Y/X_i}) \; est \; maximum, X(.) \; et \; Y(.) \; sont \; indépendants \qquad (4.7)$$

De fait, rien ne nous assure que la distribution de $G(T_{Y/X})$ sous l'hypothèse nulle soit toujours la même, [Cohen et Jensen, 1996] affirment qu'il s'agit là d'une des principales sources de sur-apprentissage en induction.

Pour mieux éclaircir notre propos, nous allons reprendre un exemple de ces auteurs. Soit deux variables a_1 et a_2 prenant leurs valeurs dans $\{1, \ldots, 6\}$. Nous présentons dans la table 4.1 tous les cas possibles de la variable aléatoire $v = \max(a_1, a_2)$.

On considère que les a_i sont indépendants et identiquement distribués, avec $P(a_i = t) = \frac{1}{6}$, et $P(a_i > t) = 1 - \frac{t}{6}$. Si nous considérons maintenant la distribution de v, nous obtenons

$$\begin{aligned} P(v > t) &= P(a_1 > t, a_2 \leq t) + P(a_1 \leq t, a_2 > t) + P(a_1 > t, a_2 > t) \\ &= (1 - \frac{t}{6})\frac{t}{6} + \frac{t}{6}(1 - \frac{t}{6}) + (1 - \frac{t}{6})(1 - \frac{t}{6}) \\ &= (1 - \frac{t}{6})(1 + \frac{t}{6}) \end{aligned}$$

	1	**2**	**3**	**4**	**5**	**6**
1	1	2	3	4	5	6
2	2	2	3	4	5	6
3	3	3	3	4	5	6
4	4	4	4	4	5	6
5	5	5	5	5	5	6
6	6	6	6	6	6	6

TAB. 4.1 – *Liste des valeurs de la variable aléatoire* $v=max(a_1,a_2)$

On remarque que

$$P(v > t) > P(a_i > t)$$

[Cohen et Jensen, 1996] observent un phénomène analogue dans l'équation 4.6, le risque critique du test $\alpha_{obs.}$ est sous-évalué, nous acceptons trop souvent l'hypothèse alternative.

Afin de remédier à cet optimisme excessif, ils préconisent l'utilisation de l'ajustement de Bonferroni conçu pour les comparaisons multiples, il répond de manière simple à la problématique de la recherche de la distribution de probabilité de la valeur maximum en modifiant la valeur du risque de première espèce rattaché au test. L'équation de définition de la région critique 4.6 devient alors

$$\Pr(\chi^2 > G(T_{Y/X})) = \alpha_{obs.}$$

$$\textit{Décider } H_1 \textit{ si } \alpha_{obs.} < 1 - (1-\alpha)^{\frac{1}{p}} \tag{4.8}$$

où p est le nombre de tests effectués i.e le nombre d'attributs candidats à la segmentation sur le noeud.

On remarque aisément que $(1-(1-\alpha)^{\frac{1}{p}} < \alpha)$, à mesure que le nombre d'attributs candidats augmente, l'acceptation d'une variable de segmentation devient difficile.

4.6 Elagage - Post-élagage

La détection de la bonne taille de l'arbre par arrêt de l'expansion a beaucoup été critiquée dans la littérature [Buntine, 1992] [Wehenkel, 1993]. Pour remédier à ses déficiences, une nouvelle méthode a été introduite par [Breiman *et al.*, 1984], elle consiste à construire un arbre aussi grand que l'on veut dans une première phase avant de le réduire pas à pas en expulsant les feuilles non-pertinentes : c'est le procédé de l'élagage. Cet dernier possède plusieurs avantages dont le premier est de ne pas nécessiter le choix toujours délicat d'un seuil comme c'était le cas dans la section précédente. D'autres avantages, peut-être moins palpables lui sont également associés :

– on sait que le taux d'erreur par resubstitution est un mauvais indicateur pour explorer

l'espace des solutions, il reste néanmoins que l'objectif final est bien sûr de construire un classifieur qui présente un taux d'erreur minimum en généralisation. L'élagage permet de post-traiter un arbre, produit avant tout pendant l'expansion pour maximiser la pureté des feuilles, en recherchant la situation qui nous rapproche le plus de notre objectif. La stratégie repose sur la capacité à procurer une meilleure estimation du taux de succès en généralisation.

– il arrive parfois que la combinaison de deux ou plusieurs attributs se révèle intéressante alors que tous les attributs pris un par un n'apportent pas significativement d'information. En adoptant un principe de construction "hurdling" i.e on accepte la segmentation sur un noeud même si elle n'est pas significative, on augmente nos chances de détecter ces interactions sans pour autant tomber dans les travers de complexité de la recherche en avant. Toutefois, l'effectivité de cet avantage dépend de plusieurs conditions : l'absence d'attributs bruités qui pourraient s'insérer dans l'arbre au bénéfice de l'acceptation à tout prix de la segmentation, et le fait que la somme de choix optimaux locaux (choix d'une variable sur un noeud à segmenter) nous amène vers un optimum global (la combinaison de m variables sur plusieurs niveaux maximisant le gain).

Il y a un très grand nombre de méthodes d'élagage en induction d'arbre [Mingers, 1989a] [Esposito *et al.*, 1995] [Knoll *et al.*, 1994] [Wehenkel, 1992]. Nous en distinguons principalement trois familles :

1. élagage en utilisant une estimation du taux d'erreur par un échantillon test;
2. élagage en utilisant une meilleure estimation du taux d'erreur sans échantillon test;
3. élagage par optimisation d'une mesure d'arbitrage entre complexité et précision

Nous reportons l'analyse de cette troisième méthode dans le chapitre suivant car il obéit à une logique un peu différente qui mérite que l'on s'y attarde plus en profondeur, d'autant plus que l'on peut l'adapter facilement à la construction de graphes, ce qui n'est pas le cas des deux premières.

4.6.1 Elagage par échantillon test

Le principe de base consiste à subdiviser l'échantillon d'apprentissage en deux parties : la première $\Omega^{a,t}$ pour l'expansion de l'arbre de taille maximum $A_{\max}$, la seconde $\Omega^{a,e}$ pour l'élagage i.e la recherche de l'arbre de plus petite taille A^* qui minimise son taux d'erreur sur cette portion. Les algorithmes diffèrent par l'exploration de l'espace des sous-arbres inclus dans $A_{\max}$.

Réduction de l'arbre par optimisation brute du taux d'erreur

Généralement, les méthodes d'élagage emploient un algorithme ascendant pour réduire les arbres, on sonde les effets de la transformation d'un noeud pré-terminal (précédent des feuilles) en feuilles. [Quinlan, 1987c] prend complètement à contre pied cet archétype en testant l'opportunité de transformation en feuille de tous les noeuds non-terminaux de l'arbre. Pour chacun d'eux, il calcule la réduction du taux d'erreur induit par la suppression de leurs sous-arbres enfants. Un noeud devient une feuille s'il maximise cette réduction (si elle est positive ou nulle bien sûr).

Bien que l'auteur ne le spécifie pas expressément, il paraît naturel en cas d'ex-aequo de supprimer le plus grand sous-arbre si l'on a une préférence pour la simplicité.

La principale critique que l'on peut adresser à cette méthode est son extrême sensibilité à l'échantillon d'élagage. Un seul individu suffit pour obtenir un arbre optimal complètement différent. Nous retombons ici dans les travers de l'optimisation en apprentissage à partir du taux d'erreur.

Réduction de l'arbre par optimisation de l'erreur calculée sur une séquence d'arbre

La méthode CART élaborée par [Breiman *et al.*, 1984] est autrement plus sophistiquée. Elle définit une série d'arbres imbriqués A_i de coûts-complexités identiques et recherche parmi ceux-ci celui qui minimise l'erreur sur l'échantillon d'élagage.

La mesure de coût-complexité se calcule de la manière suivante

$$R_\alpha(A_i) = r(A_i) + \alpha.|A_i|$$

où

- $r(A_i)$ est le taux d'erreur en resubstitution (calculé sur $\Omega^{a,t}$),
- $|A_i|$ est le nombre de feuilles dans le sous-arbre A_i,
- α est un coefficient de complexité pénalisant les arbres comportant beaucoup de feuilles.

Les séquences de sous-arbres sont définies en cherchant pour chaque valeur de α l'arbre le plus simple A_i^* qui minimise la quantité $R_\alpha(A_i)$. Lorsque α augmente, elles sont naturellement de complexité décroissante

$$|A_1^*| > |A_2^*| > \ldots > 1$$

l'arbre optimum A^* est tel que

$$A^* = \arg\max_i \varepsilon(A_i^*)$$

où $\varepsilon(A_i^*)$ est le taux d'erreur calculé sur le second échantillon $\Omega^{a,e}$.

Dans la pratique, on observe qu'autour de l'optimum, l'erreur $\varepsilon()$ reste sur un plateau assez stable alors que le nombre de feuilles varie énormément [Gueguen, 1994]. Afin d'éviter le sur-apprentissage sur le fichier d'élagage, [Breiman *et al.*, 1984] annoncent clairement leur préférence pour la simplicité en établissant la "règle d'un écart-type", on choisit l'arbre $A\sigma^*$ possédant le moins de feuilles répondant à la condition suivante

$$\varepsilon(A\sigma^*) < \varepsilon(A^*) + \sigma[\varepsilon(A^*)]$$

où $\sigma[\varepsilon(A^*)]$ est l'écart-type calculé sur une proportion, i.e

$$\sigma[\varepsilon(A^*)] = \sqrt{\frac{\varepsilon(A^*)(1-\varepsilon(A^*))}{card(\Omega^{a,e})}}$$

[Buntine et Caruana, 1991] montrent expérimentalement que cette correction réduit la taille des arbres sans modifier les performances en généralisation.

L'élagage selon CART est une procédure fiable qui fait référence en intelligence artificielle, le seul reproche que l'on peut lui adresser est sa gourmandise en observations, deux échantillons sont nécessaires pour produire un classifieur. En situation de pénurie des données, les auteurs conseillent l'utilisation de la cross-validation pour estimer la bonne taille de l'arbre, elle donne de bons résultats mais bien sûr requiert plus de temps CPU.

Il existe un perfectionnement de cette stratégie originelle, [Gelfand *et al.*, 1991] ont proposé un algorithme itératif qui construit et élague l'arbre alternativement sur les échantillons $\Omega^{a,t}$ et $\Omega^{a,e}$. Bien que la convergence vers un arbre optimal ne soit pas prouvée formellement, elle semble effective dans la pratique. Notons néanmoins, qu'il n'introduit pas d'avancée significative en terme de complexité et de performances de l'arbre.

4.6.2 Elagage par ré-estimation du taux d'erreur

A la différence de la section précédente, on n'utilise qu'un seul échantillon ici. Cela est avantageux lorsque les données sont rares et que le concept à apprendre est difficile. Le principe est toujours le même, on construit l'arbre maximum puis on essaie de le réduire en comparant un indicateur $\phi(T_Y)$ calculé sur chaque noeud pré-terminal (précédant les feuilles) et sa moyenne pondérée sur ses sommets enfants

$$\phi(T_{Y/X}) = \sum_{l=1}^{L} \frac{n_{.l}}{n} \phi[T_{Y/(X=x_l)}]$$

on décide que l'on doit transformer un noeud pré-terminal en feuille si et seulement si

$$\phi(T_Y) \leq \phi(T_{Y/X})$$

Que doit représenter la mesure ϕ? Au regard de la section précédente, elle remplace une estimation "réaliste" du taux d'erreur qui cette fois doit être calculé sur l'échantillon utilisé

pour construire l'arbre. Dans cette optique, il existe dans la littérature pléthore d'indicateurs dont la justification théorique est plus ou moins discutable [Niblett, 1987] [Cestnik *et al.*, 1987a] [Quinlan, 1987c] [Cestnik, 1990] [Quinlan, 1993a].

En réalité, le principe de base est que l'indicateur doit marquer sa sensibilité aux effectifs en préférant les groupes de plus grande taille où les estimations sont plus fiables, autrement dit en cas d'égalité du taux d'erreur par resubstitution sur un sommet et ses feuilles, la préférence doit être donnée à celui qui est estimé sur un plus grand nombre d'individus, à savoir le noeud pré-terminal. Nous retrouvons dans cette formulation, bien qu'elle ne soit nulle part explicite dans les travaux des auteurs concernés, la propriété de fusion que nous avons introduite dans le chapitre sur les mesures d'évaluation de la segmentation.

Nous allons vérifier cette propriété sur deux indicateurs exposés dans [Mingers, 1989a] en les illustrant sur le cas très particulier d'un noeud portant deux feuilles de distributions de classes identiques dont voici le tableau de contingence associé

	gauche	droite	$\sum$
$Y = y_1$	$\frac{n_{1.}}{2}$	$\frac{n_{1.}}{2}$	$n_{1.}$
$Y = y_2$	$\frac{n_{2.}}{2}$	$\frac{n_{2.}}{2}$	$n_{2.}$
$\sum$	$\frac{n}{2}$	$\frac{n}{2}$	n

On fixe par hypothèse $n_{1.} > n_{2.}$, la classe majoritaire y_1 est conclusion.

Le taux d'erreur espéré

[Cestnik *et al.*, 1987a] propose un indicateur qu'ils présentent comme l'estimateur du taux d'erreur en généralisation. Il n'est pas sans rappeler l'estimateur laplacien du taux d'erreur, ici la formulation diffère quelque peu

$$e(T_Y) = \frac{n - n_c + K - 1}{n + K}$$

avec n_c l'effectif de la classe conclusion.

Appliqué sur notre noeud, le taux calculé est

$$e(T_Y) = \frac{n_{2.} + 1}{n + 2}$$

Sur les feuilles, nous obtenons

$$\begin{aligned} e(T_{Y/X}) &= 2\frac{n/2}{n}\left(\frac{n_{2.}/2 + 1}{n/2 + 2}\right) \\ &= \frac{n_{2.} + 2}{n + 4} \end{aligned}$$

On constate aisément alors que $\Delta = e(T_Y) - e(T_{Y/X})$ est positif, impliquant ainsi la transformation du noeud en feuille. La même vérification peut être réalisée sur la seconde mesure que nous présentons ci-après.

Le taux d'erreur pessimiste

[Quinlan, 1987c] a proposé un indicateur, le taux d'erreur pessimiste, dont il donnera plus tard un "habillage" statistique [Quinlan, 1993a] sans changer le principe initial : rajouter au taux d'erreur en resubstitution sur les feuilles une fraction de sa variance qui est fonction décroissante du nombre d'observations intervenant dans le calcul, cela afin de pénaliser les petits effectifs.

Il modifie légèrement la définition du taux d'erreur en incorporant la correction de continuité que l'on utilise généralement lorsque l'on veut en statistique approcher une distribution discrète, en l'occurrence la loi binomiale, à l'aide d'une loi continue, que l'on expecte normale dans le cas présent.

$$rc(T_Y) = \frac{n - n_c + 0.5}{n}$$

Sur les feuilles, le taux d'erreur moyen est calculé à l'aide de la formule suivante

$$\begin{aligned} rc(T_{Y/X}) &= \frac{\sum_l [r(T_{Y/(X=x_l)}) + 0.5]}{\sum_l n_{.l}} \\ &= \frac{\sum_l r(T_{Y/(X=x_l)}) + \frac{L}{2}}{\sum_l n_{.l}} \end{aligned}$$

dont on peut calculer l'écart-type

$$\sigma_{rc} = \sqrt{\frac{rc(T_{Y/X})[1 - rc(T_{Y/X})]}{n}}$$

La règle de décision amenant la suppression des feuilles s'écrit

$$rc(T_Y) < rc(T_{Y/X}) + \beta \times \sigma_{rc}$$

où β est un paramètre positif que l'on peut faire varier à loisir pour pénaliser (β grand) ou favoriser (β faible) les arbres de grande taille. Dans la plupart des cas, l'auteur conseille la valeur $\beta = 1$.

Vérifions de nouveau la propriété de fusion sur notre exemple. L'erreur initiale est

$$rc(T_Y) = \frac{n_{2.} + 0.5}{n}$$

sur les feuilles, elle devient

$$rc(T_{Y/X}) = \frac{n_{2.} + 1}{n}$$

Sachant que $\sigma_{rc} > 0$, on constate que la correction de continuité suffit pour décider la suppression des sommets enfants.

Ces calculs sur un petit exemple très particulier n'ont bien sûr pas valeur de démonstration, ils illustrent tout simplement le fait que derrière des formulations très rigoureuses (§ 4.6.2) ou des

recettes de cuisine fondées sur le bon sens (§ 4.6.2.0) se cachent en réalité les principes fondateurs que sont la pénalisation des petits effectifs pour l'estimation des probabilités, et la préférence pour la simplicité à performances égales, que nous pouvons rapprocher avec le principe de fusion pour les mesures d'évaluations des segmentations.

4.7 Conclusion

Le principal enseignement que l'on peut tirer de ce chapitre est qu'il n'existe pas de procédure automatisée qui constitue la solution optimale quel que soit le domaine d'étude. La préférence à la simplicité n'est pas une panacée.

En revanche, compte tenu des connaissances dont nous disposons sur le problème, nous pouvons choisir la meilleure stratégie. Et mieux même, dans une phase exploratoire sur des données, sachant que la plupart des exemples connus montrent une adéquation des modèles très simples, et aussi parce que l'on peut penser raisonnablement qu'il y a certainement des attributs non-pertinents dans la base qui n'a pas été a priori optimisée pour l'apprentissage, la préférence à la simplicité est une bonne stratégie. Cela est d'autant plus vrai qu'il y a peu de chances que le concept à apprendre corresponde effectivement à une fonction booléenne, dans ce cas d'inadaptation du système de représentation [Dietterich et Kong, 1995b] affirme que le biais sera toujours élevé et de toute manière incompressible, dès lors il importe de travailler sur la variance pour réduire l'erreur globale.

Les heuristiques exposées dans ce chapitre reposent sur des considérations plus ou moins discutables. On sait qu'elles nous mènent d'une certaine manière vers une minimisation de l'erreur en recherchant le meilleur compromis entre le biais et la variance de l'erreur. Mais le paramètrage optimal dépend de notre intuition et de notre connaissance du domaine, la meilleure attitude est de procéder à plusieurs tentatives pour déterminer le réglage optimal pour une base de données.

Enfin, en ce qui concerne le débat sur les avantages et inconvénients de la règle d'arrêt face à l'élagage, nous différerons la réponse à cette question dans le chapitre suivant. En effet, il nous semble peu approprié de comparer des procédures qui s'appuyent sur des critères et des heuristiques différentes.

Chapitre 5

Détermination de la taille optimale du graphe d'induction (suite)

5.1 Introduction

Dans le chapitre précédent, nous nous sommes intéressés à des heuristiques, plus ou moins heureuses, qui nous permettaient de rechercher la "bonne taille de l'arbre" et d'éviter le partitionnement excessif. Ce défaut est en grande partie imputable à la mesure d'évaluation utilisée. En effet, la propriété d'indépendance des mesures à base d'entropie [Aczel et Daroczy, 1975] implique qu'une optimisation locale revient à une optimisation globale [Breiman *et al.*, 1984]. De fait, par la propriété de minimalité, ces indicateurs privilégieront toujours les arbres les plus purs, à la limite un individu par feuille nous donnera une entropie nulle.

Pour remédier à ce problème, il nous faut un indicateur de crédibilité globale qui nous permet de soupeser le bénéfice du passage d'une partition à une autre i.e des graphes de tailles différentes [Sorkin, 1983] [Rendell, 1986]. De fait, le problème de l'induction sera ramené à un problème d'optimisation, l'hypothèse sous-jacente étant que le graphe qui minimise la mesure est celui qui se comportera le mieux en généralisation. Le bien fondé de cette extrapolation peut être discutable, tout dépend en réalité de la justification et de l'interprétation de la mesure proposée.

Nous recensons principalement deux stratégies qui répondent de manière adéquate à cette formulation de l'induction :

1. la description minimale des données [Rissanen, 1978] : elle recherche le meilleur compromis entre les coûts de description du modèle et de ses exceptions. Le classifieur est ainsi analysé sous sa forme structurelle, son codage a d'ailleurs été l'objet de vives polémiques [Quinlan et Rivest, 1989] [Wallace et Patrick, 1993]. Cette approche constitue une interprétation élégante de la problématique bayésienne où la justification des distributions a priori est toujours délicate [Buntine, 1992].

2. analyse du tableau de contingence associé à la partition : l'objectif est de rechercher la meilleure partition qui maximise une mesure d'incertitude, répondant à la propriété de fusion, calculée sur le tableau de contingence [Zighed *et al.*, 1992] [Rauber et Steiger-Garccedillo, 1993]. L'absence de biais en faveur des partitions fines permet d'éviter le phénomène du sur-apprentissage. Nous nous intéresserons plus particulièrement à une interprétation originale permettant de rattacher l'induction par graphes à la problématique générale de la régression [Goodman et Kruskall, 1954] [Light et Margolin, 1971].

Ces deux stratégies présentent l'avantage de s'appliquer sans modifications à l'élaboration des arbres et des graphes, en tant que telles nous les considérons comme des axes privilégiés d'une approche unifiée. En l'absence de toute contrainte sur la forme du modèle, rien n'empêchera la méthode de produire des arbres si telle est la structure qui optimise la mesure que l'on s'est choisie.

Enfin, puisque cette nouvelle approche ramène la problématique de l'induction à une minimisation sans contrainte d'un indicateur, elle ouvre un nouveau champ de recherche fertile pour dépasser les limites de la construction gloutonne [Meisel et Michalopoulos, 1973] [Garey et Graham, 1974] en adoptant des méthodes d'exploration plus élaborées comme le recuit simulé, la recherche en avant limitée, les algorithmes génétiques... [Koza, 1991] [Bucy et Diesposti, 1993] [Weiss et Indurkhya, 1993]

Dans ce chapitre, nous présenterons tour à tour les deux approches d'évaluation globale des graphes ci-dessus en mettant en avant les arguments théoriques qui justifient l'hypothèse de bon comportement en généralisation des classifieurs produits. Nous procéderons par la suite à une expérimentation où l'on essaiera de répondre à la question de savoir si l'élagage permet, pour les raisons invoquées dans le chapitre précédent, de trouver véritablement de meilleures partitions que la stratégie de la règle d'arrêt dans la construction des arbres.

5.2 L'induction de graphes par la théorie de la description minimale des messages

5.2.1 L'inférence inductive par encodage minimum

La théorie de la description minimale des messages[17] [Rissanen, 1978] est un cadre global permettant de choisir un modèle optimal dans l'espace des hypothèses. Elle est très similaire des travaux de [Wallace et Boulton, 1968] sur la longueur minimale des messages[18] dont elle emprunte d'ailleurs plusieurs résultats [Baxter et Oliver, 1994]. On les confond très souvent dans la littérature car leur postulat de base est bien le même : l'inférence inductive par le codage minimum de description des données.

17. Minimum description length (MDL)
18. Minimum message length (MML)

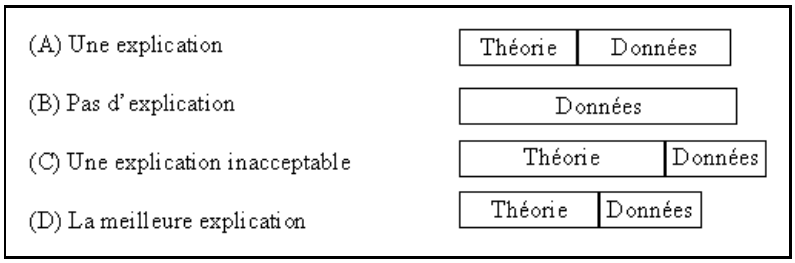

FIG. 5.1 – *Réduction de la description des données par une théorie*

Le principe initial est relativement simple. Soit Ω^a un ensemble de données quelconque que l'on doit expliquer à l'aide d'une théorie. Dans notre cas, la théorie sera un graphe d'induction, elle décrit un ensemble de chemins en y associant une classe de la variable à prédire. L'inférence par minimisation de la description minimale (resp. de la longueur du message) consiste alors à chercher le classifieur qui minimise la somme de la description de la théorie et de la description des exceptions à cette théorie (resp. des individus sachant cette théorie, figure 5.1 [Oliver *et al.*, 1992]). La théorie est rejetée si cette somme excède le coût de la description de tous les individus. Ce principe peut être utilisé dans des domaines divers comme la classification non-supervisée [Wallace et Dowe, 1994], l'estimation de paramètres dans la régression [Baxter et Dowe, 1994] ou en statistique mathématique, en fait à chaque fois que l'on a à choisir un modèle parmi un ensemble d'hypothèses.

Complexité de Kolmogorov

L'encodage minimum répond en fait à une formalisation théorique de l'inférence inductive par la "complexité algorithmique" développée par [Solomonoff, 1964], [Kolmogorov, 1965] et [Chaitin, 1966]. Elle est fondée sur la notion de "complexité de Kolmogorov" qui définit une mesure de complexité universelle pour des chaînes binaires.

Soit B une machine quelconque, la complexité de Kolmogorov pour une chaîne s sur la machine B sera la longueur $K_B(s)$ de la plus petite chaîne en entrée (le programme M) telle que B produira la chaîne s. Si B est une machine de Turing universelle, alors les auteurs pré-cités ont montré que la complexité de Kolmogorov $K(s)$ sera telle que

$$K(s) \leq K_B(s) + k$$

pour toutes les chaînes binaires s, k est une constante qui ne dépend que de B.

Le programme M peut être un modèle qui explique la chaîne en sortie s, la distribution de probabilité de cette dernière est déterminée par Solomonoff-Levin [Oliveira, 1994]

$$P(s) = \sum_{M:B(M)=s} 2^{-|M|}$$

où $|M|$ est la longueur du programme. Dans la pratique, elle n'est pas calculable puisque nous ne pouvons définir tous les programmes qui mènent à s. En revanche, il existe une relation déterministe entre cette probabilité et la complexité de Kolmogorov

$$P(s) = e^{-K(s)+O(1)}$$

De fait, nous observons deux résultats importants :

– plus le modèle est complexe, moins il aura de chance d'apparaître;

– les modèles de complexité identique sont équiprobables.

Nous retrouvons exactement les propriétés de la distribution a priori des modèles utilisée par [Wehenkel, 1992] pour composer sa spécification bayésienne de construction des arbres de décision.

Une justification bayésienne

La formulation bayésienne permet de justifier l'inférence par encodage minimum. Rappelons que la probabilité a posteriori d'obtenir un modèle connaissant les données s'écrit

$$P(M/\Omega^a) = \frac{P(\Omega^a/M)P(M)}{P(\Omega^a)}$$

Maximiser $P(M/\Omega^a)$ revient à minimiser $-\log(P(M/\Omega^a))$, donc nous chercherons le modèle M^* tel que

$$M^* = \arg\min_M [-\log(P(M)) - \log(P(\Omega^a/M))] \tag{5.1}$$

Si nous arrivons à trouver une manière optimale d'encoder les graphes comme des chaînes binaires, ou du moins calculer la quantité d'information nécessaire pour le faire, il paraît alors logique d'utiliser une version légèrement modifiée de la distribution de Solomonoff-Levin

$$P(H) = 2^{-|H|}$$

et transformer ainsi l'équation 5.1 en

$$M^* = \arg\min_M [K(M) + K(\Omega^a/M)] \tag{5.2}$$

qui correspond à l'équation de base de la théorie de la description minimale des données.

Numéro	Y	X_1	X_2	X_3
1	Y_1	1	2	1
2	Y_1	1	2	1
3	Y_1	2	1	1
4	Y_1	2	1	1
5	Y_1	2	1	1
6	Y_2	2	2	2
7	Y_2	1	2	2
8	Y_2	1	1	1
9	Y_2	1	1	1
10	Y_2	1	1	1

TAB. 5.1 – *Fichier exemple*

5.2.2 Codage d'un arbre de décision

L'efficacité des MDL et MML dépend énormément de la capacité des chercheurs à trouver des estimations satisfaisantes des complexités de Kolmogorov du classifieur et des données afférentes i.e produire un codage optimal pour les décrire. Il en existe de très nombreuses variantes : pour les arbres, dans l'induction [Wallace et Patrick, 1993] ou spécifiquement pour l'élagage [Mehta *et al.*, 1995] [Utgoff, 1995]; pour les graphes [Oliver et Wallace, 1991] [Oliveira et Sangiovanni-Vincentelli, 1995].

Nous reprenons dans ce qui suit le codage sur arbre binaire, pour un problème à deux classes, proposé par [Quinlan et Rivest, 1989][19], qui historiquement sont les premiers à avoir adopté cette approche pour la classification supervisée, nous l'illustrerons par le fichier exemple suivant (Table 5.1).

Encodage d'une chaîne binaire finie

[Quinlan et Rivest, 1989] proposent une technique assez particulière pour calculer le coût de transmission d'une chaîne binaire où il y a n éléments avec une classe majoritaire (y_k) et ce exceptions ($ce = n - n_{k.}$, nous pouvons aussi les qualifier de contre-exemples). Nous savons que si cette classe est majoritaire, la borne haute du nombre d'exceptions est égale à $b = \frac{n}{2}$ si n est pair ($b = \frac{n-1}{2}$ sinon).

19. [Wallace et Patrick, 1993] ont relevé que la distribution de probabilité des chaînes binaires utilisée par ces auteurs n'est pas appropriée. Fort heureusement, l'erreur se traduit par un coefficient multiplicatif ($\times 2$) appliqué sur le coût total de description, l'optimisation porte dès lors sur le double de la bonne valeur, ce qui ne change en rien le choix du meilleur arbre dans l'induction. Nous avons ôté ce coefficient dans notre présentation.

Dans la transmission du message, nous devons donc spécifier tour à tour :

– la quantité ce, cela requiert

$$\log(b+1)\ \ bits$$

– sachant ce, dénombrer les cas possibles de dispositions de la chaîne, cela requiert

$$\log(nce)\ \ bits$$

[Quinlan et Rivest, 1989] pensent que ce coût est justifié dans la mesure où la disposition des individus dans la base de données n'est pas fixe, toutes les permutations possibles peuvent intervenir sans que cela ne doive changer le résultat de l'induction.

Prenons un exemple simple, si la chaîne comporte 4 éléments et 1 exception à la classe Y_1, les cas possibles seront

$$Y_2Y_1Y_1Y_1$$
$$Y_1Y_2Y_1Y_1$$
$$Y_1Y_1Y_2Y_1$$
$$Y_1Y_1Y_1Y_2$$

De fait, le coût total de transmission devient

$$L(n, ce, b) = \log(b+1) + \log(nce)$$

Si nous l'appliquons à la table 5.1, le coût de description de tous les individus si l'on pose arbitrairement que la classe 1 est majoritaire (on aurait bien pu décider dans ce cas précis que c'était la classe 2, cela n'aurait changé en rien aux résultats) est

$$\begin{aligned} Description\_Initiale &= \log(6) + \log(252) \\ &= 10.562\ \ bits \end{aligned}$$

Encodage de l'arbre

Il faut trouver un système d'encodage qui soit d'autant plus court que l'arbre est de petite taille. Les auteurs proposent un système de description récursif qui parcourt l'arbre de la racine vers les feuilles. On distingue les feuilles des noeuds :

– une feuille est codée "0", elle est suivie par le code de la classe par défaut;

– un noeud est codé "1", suivi par la description de l'attribut qui est mis en oeuvre, puis l'on décrit successivement les noeuds enfants suivant les branches de l'arbre, de la gauche vers la droite.

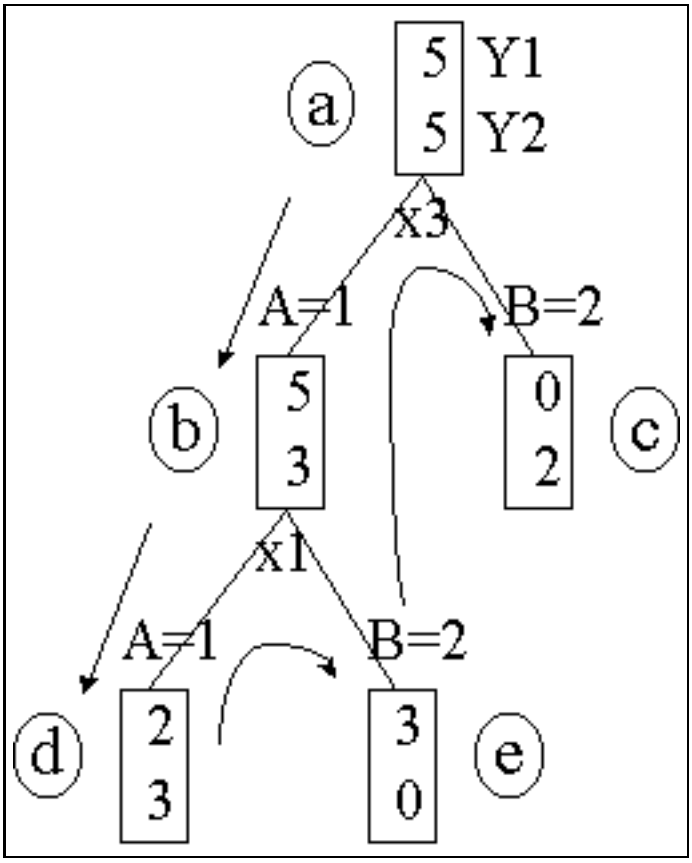

FIG. 5.2 – *Un arbre construit sur les données de la table 5.1*

Pour illustrer cette technique, nous allons l'appliquer sur un arbre (figure 5.2) construit sur notre fichier (table 5.1), dans l'ordre (a, b, d, e, c) nous obtiendrons

$$1\ x_3\ 1\ x_1\ 0\ \ Y_2\ 0\ Y_1\ 0\ Y_2$$

dont le coût de transmission sera

$$\begin{aligned} Théorie &= 1 + \log(3) + 1 + \log(2) + 1 + 1 + 1 + 1 + 1 + 1 \\ &= 10.585\ bits \end{aligned}$$

Le coût pour décrire l'intervention de l'attribut "x_3" dans l'arbre est égal au nombre de bits nécessaires pour la spécifier dans la base de départ, comme nous disposons de 3 variables, la valeur adéquate sera log(3). En revanche, sachant que la variable "x_3" a déjà été utilisée plus haut, il ne faut qu'un bit pour décrire l'introduction de la variable "x_2".

Encodage des exceptions

Pour coder les exceptions sur les feuilles, nous suivons le même principe que celui décrit dans l'encodage des chaînes binaires, nous avons la classe par défaut (la classe majoritaire), nous devons spécifier le nombre d'exceptions et décrire tous les individus correspondants.

Dans notre exemple, nous obtiendrons, toujours de la gauche vers la droite :

$$\begin{aligned} Exceptions &= L(5, 2, 3) + L(3, 0, 1) + L(2, 0, 1) \\ &= 7.322\ bits \end{aligned}$$

L'objectif de l'induction revient ainsi à rechercher l'arbre qui minimise la somme ($Théorie + Exceptions$), la partition sera acceptée si elle est inférieure à la description initiale de tous les individus. En ce sens la problématique de l'inférence est très similaire de celle de la compression de l'information.

L'adaptation de ce système à des attributs non-binaires, et à plus de deux classes est assez périlleuse, et fait d'ailleurs l'objet de débats passionnés. Conscients de l'influence néfaste que peut avoir un codage non optimal de l'arbre, notamment l'occurrence de codes redondants, sur les capacités de l'algorithme à trouver les "bonnes" formes correspondant au concept à apprendre, [Quinlan, 1993a] propose une variante où l'on essaie de minimiser

$$w \times Théorie + Exceptions \tag{5.3}$$

Le paramètre w qui varie entre 0 et 1, indiquerait la préférence de l'expert à la simplicité. Quand w est très faible, on cherche le modèle qui colle le mieux aux données (on se rapproche de la stratégie originelle d'ID3 [Quinlan, 1979]); dans le cas contraire, on préférerait les modèles les moins complexes.

Dans la pratique, la théorie de la description minimale (MDL et MML) se révèle robuste et souple en induction, l'applicabilité sur des systèmes de représentations différents dépend tout simplement de la capacité à trouver un codage optimal de la théorie. Ainsi, [Quinlan, 1993a] l'utilise avec succès dans la réduction des bases de règles issues des arbres comme nous le verrons plus loin dans cette thèse.

5.3 L'induction de graphes par régression

A ses débuts, les arbres de décision en statistique étaient destinés avant tout à la régression par arbre, où la variable à prédire était quantitative [Morgan et Sonquist, 1963]. La spécification était alors proche de l'analyse de variance [Breiman *et al.*, 1984]. Au cours de nos recherches, nous nous sommes demandés s'il était possible de retrouver une telle spécification sur une endogène qualitative. Nous avons principalement utilisé les résultats théoriques de [Light et Margolin, 1971] sur le gain relatif d'entropie quadratique (le τ de [Goodman et Kruskall, 1954]), nous présentons cette approche principalement dans [Rakotomalala et Zighed, 1997].

5.3.1 Une réinterprétation de la variance sur données qualitatives

La variance est très peu appliquée sur les variables qualitatives nominales parce qu'elle est souvent comprise comme la somme pondérée des écarts quadratiques à la moyenne, et cette dernière notion n'existe pas sur ce type de variable. C'est une erreur, [Gini, 1938] a montré qu'en utilisant une formulation différente, il était possible de la calculer en introduisant un indicateur de différence approprié.

Selon [Gini, 1938], la somme des carrés des écarts à la moyenne[20] est équivalente à la somme des écarts au carré deux à deux de l'échantillon, il y a $n2$ cas possibles :

$$TSS = \sum_{i=1}^{n}(y_i - \overline{y})^2 \tag{5.4}$$

$$TSS = \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}(y_i - y_j)^2 \tag{5.5}$$

Pour les variables nominales, nous pouvons définir l'opérateur de différence de la manière suivante [Light et Margolin, 1971]

$$d_{ij} = \begin{cases} 0 \ si \ x_i = x_j \\ 1 \ si \ x_i \neq x_j \end{cases}$$

L'équation 5.5 devient

$$TSS = \frac{1}{2n}\sum_{i}\sum_{j} d_{ij}^2$$

20. Total sum of square

sachant que d_{ij} prend ses valeurs dans $\{0,1\}$, nous pouvons écrire

$$TSS = \frac{1}{2n} \sum_i \sum_j d_{ij}$$

Il s'agit donc tout simplement de compter le nombre de fois où chaque individu peut prendre une modalité différente des observations composant le reste de l'échantillon. Pour fixer les idées, prenons le cas du premier individu portant l'étiquette k. Le nombre de fois où $d_{1j} = 1$ pour $(j = 1, \ldots, n)$ est égal à $(n - n_{k.})$, où $n_{k.} = card(\{\omega \in \Omega^a \,/\, Y(\omega) = k\})$. Si l'on fait la somme des cas possibles où $d_{ij} = 1$, pour tous les individus portant y_k, nous obtenons[21]

$$\sum_{i=1}^{n} (y_i = k) \times (n - n_{k.}) = \sum_{m=1}^{n_{k.}} (n - n_{k.}) = n_{k.}(n - n_{k.})$$

Dès lors, si l'on considère l'ensemble des modalités prises par la variable $Y(.)$, il vient

$$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} = \sum_{k=1}^{K} n_{k.}(n - n_{k.})$$

et

$$TSS = \frac{n}{2} \sum_{k=1}^{K} n_{k.}(n - n_{k.})$$

et la variance devient tout simplement

$$\begin{aligned} V(Y) &= \frac{1}{n} TSS \\ V(Y) &= \frac{1}{2} \sum_{k=1}^{K} n_{k.}(n - n_{k.}) \end{aligned}$$

Nous retrouvons ici une forme très proche, à un facteur multiplicatif près, d'entropie quadratique qui est utilisée dans CART [Breiman *et al.*, 1984]. Notons que cette formulation est autrement plus élégante que celle préconisée par [Heath *et al.*, 1993b], qui pour retrouver une interprétation en terme de variance, affecte un code numérique arbitraire à chaque modalité de $Y(.)$ et calcule $V(Y)$ à l'aide de l'équation 5.4.

5.3.2 Calcul du coefficient de régression sur données qualitatives

On peut représenter un graphe à l'aide d'un tableau de contingence dans lequel nous recensons la distribution de classes selon les feuilles. Les segmentations et fusions successives entraînent une complexité plus ou moins croissante, en nombre de colonnes de ce tableau.

21. $(y_i = k)$ est une variable indicatrice égale à

$$\begin{cases} 1, \; si \; y_i = k \\ 0, \; si \; y_i \neq k \end{cases}$$

Nous pouvons caractériser le problème d'induction par la recherche de la variation de la précision de la prédiction de la classe d'un individu dans deux cas distincts :

- on ne dispose d'aucune information mise à part la distribution des classes dans l'échantillon total (la marge du tableau de contingence);
- on connaît la feuille du graphe dans laquelle il se trouve (une des colonnes du tableau de contingence).

On peut dès lors considérer que les feuilles du graphe sont autant de modalités prises par un attribut fictif $X'(.)$, notre propos revient alors à chercher l'attribut $X'_*(.)$ qui maximise sa liaison avec $Y(.)$.

A l'instar de la problématique générale de la régression, cette liaison peut être qualifiée à l'aide d'un coefficient de corrélation que l'on calcule sur des données nominales en mettant en oeuvre la définition de la variance produite ci-dessus. En effet, par analogie avec la TSS, la somme des carrés des écart intra-classes[22] i.e expliquée par la connaissance de $X'(.)$ s'écrit :

$$WSS = \sum_{l=1}^{L} \frac{n_{.l}}{2} \sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}} (1 - \frac{n_{kl}}{n_{.l}}) \tag{5.6}$$

La définition du coefficient de corrélation sur variables catégorielles devient ainsi

$$R^2 = \frac{TSS - WSS}{WSS} \tag{5.7}$$

Nous remarquons au numérateur une quantité qui correspond à la somme des carrés des écarts inter-classes[23].

Ce schéma a permis à [Light et Margolin, 1971] d'extraire une loi de distribution statistique sous l'hypothèse nulle d'indépendance des caractères $Y(.)$ et $X'(.)$. La quantité

$$c = (n-1)(K-1)R^2 \tag{5.8}$$

suit une loi de χ^2 à $(K-1)(L-1)$ degrés de liberté.

Nous adopterons comme mesure de qualité du graphe le niveau de significativité du test de liaison calculé sur le tableau de contingence décrivant la partition globale de l'espace de représentation

$$p_c = P(\chi^2_{(K-1)(L-1)} > c) \tag{5.9}$$

Tout accroissement de la complexité du classifieur, avec une évolution croissante du nombre de noeuds terminaux L, est pénalisé par une augmentation du nombre de degrés de liberté du test. Cet indicateur étant une probabilité, il varie entre 0 et 1. Avec cette interprétation en

22. Within sum of squares
23. Between sum of squares

terme d'analyse de variance sur variables catégorielles [24], nous nous affranchissons de l'estimation délicate de la distribution des concepts dans l'espace des hypothèses.

Dans cette approche, l'induction par graphes est alors justifiée par la recherche du super-attribut $X'_*(.)$ dont le coefficient de corrélation avec la variable à prédire $Y(.)$ est maximum, autrement dit qui s'écarte le plus de l'hypothèse nulle d'indépendance. Nous retrouvons une formulation analogue à celle de la régression dans un espace totalement continu.

5.3.3 Propriétés de la mesure globale d'évaluation des partitions

Nous avions postulé au départ que l'indicateur de qualité du graphe devait assurer au mieux l'arbitrage entre complexité et précision. Nous montrons dans ce paragraphe qu'en tant que mesure d'incertitude sur la partition globale, il répond parfaitement aux propriétés mises en avant par Zighed [Zighed, 1985], notamment en ce qui concerne la sensibilité aux effectifs et la fusion.

Par souci de cohérence avec nos notations dans la présentation de l'entropie quadratique dans le chapitre sur les mesures, nous utiliserons ici une formule alternative de R^2

$$R^2 = \frac{\widehat{E}(T_Y) - \widehat{E}(T_{Y/X})}{\widehat{E}(T_Y)}$$

où

$$\begin{aligned} \widehat{E}(T_Y) &= \sum_{k=1}^{K} \frac{n_{k.}}{n}(1 - \frac{n_{k.}}{n}) \\ \widehat{E}(T_{Y/X}) &= \sum_{l=1}^{L} \frac{n_{.l}}{n} \sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}}(1 - \frac{n_{kl}}{n_{.l}}) \end{aligned}$$

Proposition 6 *Symétrie*

Preuve:

Les indicateurs étant des opérateurs linéaires, la commutativité de l'addition résout la démonstration.

Proposition 7 *Maximalité*

Soit une partition $S = \{S_1, ..., S_L\}$, si $\forall l \in \{1, ..., L\}, n_{1l} = ... = n_{Kl}$ alors $p_c(S)$ est maximale.

Preuve:

Si la distribution des classes est la même dans tous les sommets terminaux, cela veut dire qu'il en est de même dans le sommet initial. En effet, $n_{k.} = \sum_{l=1}^{L} n_{kl}$, et $n_{1.} = ... = n_{K.}$. Dès lors,

24. CATANOVA: Categorical analysis of variance

$\widehat{E}(T_Y) = \widehat{E}(T_{Y/X})$ et $c = 0$. Notre indicateur étant un niveau de signification, quels que soient $K > 1$ et $L > 1$, $P(\chi^2_{(L-1)(K-1)} > 0) = 1$.

Proposition 8 *Minimalité*

Soit une partition $S = \{S_1, ..., S_L\}$, si $\forall l \in \{1, ..., L\}, \exists!\, k \in \{1, ..., K\}$ tel que $n_{kl} = n_{.l}$ alors $p_c(S)$ est minimale.

Preuve:

Lorsque $n_{kl} = n_{.l}$, $n_{jl} = 0$ $(j \neq k)$. Cela implique $\sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}}(1 - \frac{n_{kl}}{n_{.l}}) = 0$, donc

$$\begin{aligned} \widehat{E}(T_{Y/X}) &= \sum_{l=1}^{L} \frac{n_{.l}}{n} \sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}}(1 - \frac{n_{kl}}{n_{.l}}) \\ &= \sum_{l=1}^{L} \frac{n_{.l}}{n} \times 0 \\ &= 0 \end{aligned}$$

Dès lors,

$$\begin{aligned} R^2 &= \frac{\widehat{E}(T_Y) - \widehat{E}(T_{Y/X})}{\widehat{E}(T_Y)} = \frac{\widehat{E}(T_Y)}{\widehat{E}(T_Y)} \\ &= 1 \end{aligned}$$

On sait que $0 \leq R^2 \leq 1$, il ne peut pas exister une autre partition S' tel que $R^{2(\prime)} > 1$. Il est donc évident que c est maximum, et que $P(\chi^2_{(L-1)(K-1)} > c)$ est minimum.

Proposition 9 *Sensibilité à la taille de l'effectif*

Soit une partition $S = \{S_1, ..., S_L\}$, si l'on multiplie tous les effectifs par une quantité m, $S' = \{m * S_1, ..., m * S_L\}$ avec $m > 1$, alors $p_c(S) > p_c(S')$.

Preuve:

Montrons tout d'abord qu'une augmentation proportionnelle des effectifs ne modifie en rien l'indice de Gini.

$$\widehat{E}(T_Y) = \sum_{k=1}^{K} \frac{n_{k.}}{n}(1 - \frac{n_{k.}}{n})$$

En multipliant tous les effectifs par une quantité m, $\widehat{p}_{k.} = \frac{m*n_{k.}}{m*n} = \frac{n_{k.}}{n}$. Ainsi $\widehat{E}(T'_Y) = \widehat{E}(T_Y)$.

De la même manière $\widehat{E}(T'_{Y/X}) = \widehat{E}(T_{Y/X})$.

Partant de ces deux égalités, il vient

$$\frac{\widehat{E}(T'_Y) - \widehat{E}(T'_{Y/X})}{\widehat{E}(T'_Y)} = \frac{\widehat{E}(T_Y) - \widehat{E}(T_{Y/X})}{\widehat{E}(T_Y)}$$

puisque $m > 1$,

$$(n-1) \times (K-1) \times \frac{\widehat{E}(T_Y) - \widehat{E}(T_{Y/X})}{\widehat{E}(T_Y)} < (n \times m - 1) \times (K-1) \times \frac{\widehat{E}(T'_Y) - \widehat{E}(T'_{Y/X})}{\widehat{E}(T'_Y)}$$

$$c < c'$$

sachant que $c \rightsquigarrow \chi^2_{(L-1)(K-1)}$ et $c' \rightsquigarrow \chi^2_{(L-1)(K-1)}$,

$$P(\chi^2_{(L-1)(K-1)} > c) > P(\chi^2_{(L-1)(K-1)} > c')$$

$$p_c(S) > p_c(S')$$

Proposition 10 *Fusion*

Soit une partition $S = \{S_1, ..., S_i, ..., S_j, ..., S_L\}$, s'il existe un doublet i et j tel que $S_i = m * S_j$ i.e la distribution des classes dans deux sommets est proportionnelle, alors la partition fusionnée $S = \{S_1, ..., S_i + S_j, ..., S_L\}$ est telle que $p_c(S) > p_c(S')$.

Preuve:

La distribution initiale des classes est la même, donc $\widehat{E}(T'_Y) = \widehat{E}(T_Y)$. Vérifions si l'égalité est respectée pour $\widehat{E}(T_{Y/X})$.

$$\begin{aligned}
\widehat{E}(T_{Y/X}) &= \sum_{l=1}^{L} \frac{n_{.l}}{n} \sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}} (1 - \frac{n_{kl}}{n_{.l}}) \\
&= ... + \frac{n_{.i}}{n} \sum_{k=1}^{K} \frac{n_{ki}}{n_{.i}} (1 - \frac{n_{ki}}{n_{.i}}) + ... + \frac{n_{.j}}{n} \sum_{k=1}^{K} \frac{n_{kj}}{n_{.j}} (1 - \frac{n_{kj}}{n_{.j}}) + ...
\end{aligned}$$

Puisque $n_{ki} = m \times n_{kj}$, $n_{.i} = \sum_k n_{ki} = \sum_k m \times n_{kj} = m \times n_{.j}$

$$\begin{aligned}
\sum_{k=1}^{K} \frac{n_{ki} + n_{kj}}{n_{.i} + n_{.j}} (1 - \frac{n_{ki} + n_{kj}}{n_{.i} + n_{.j}}) &= \sum_{k=1}^{K} \frac{(1+m) n_{kj}}{(1+m) n_{.j}} (1 - \frac{(1+m) n_{kj}}{(1+m) n_{.j}}) \\
&= \sum_{k=1}^{K} \frac{n_{kj}}{n_{.j}} (1 - \frac{n_{kj}}{n_{.j}})
\end{aligned}$$

Ainsi,

$$\begin{aligned}
\widehat{E}(T_{Y/X}) &= ... + (\frac{n_{.i} + n_{.j}}{n}) \sum_{k=1}^{K} \frac{n_{ki} + n_{kj}}{n_{.i} + n_{.j}} (1 - \frac{n_{ki} + n_{kj}}{n_{.i} + n_{.j}}) + ... \\
\widehat{E}(T_{Y/X}) &= \widehat{E}(T'_{Y/X})
\end{aligned}$$

Comme précédemment, il vient encore une fois :

$$\begin{aligned} \frac{\widehat{E}(T_Y) - \widehat{E}(T_{Y/X})}{\widehat{E}(T_Y)} &= \frac{\widehat{E}(T'_Y) - \widehat{E}(T'_{Y/X})}{\widehat{E}(T'_Y)} \\ (n-1)*(K-1)*\frac{\widehat{E}(T_Y) - \widehat{E}(T_{Y/X})}{\widehat{E}(T_Y)} &= (n-1)*(K-1)*\frac{\widehat{E}(T'_Y) - \widehat{E}(T'_{Y/X})}{\widehat{E}(T'_Y)} \\ c &= c' \end{aligned}$$

Mais ici $c \rightsquigarrow \chi^2_{(L-1)(K-1)}$ et $c' \rightsquigarrow \chi^2_{(L-2)(K-1)}$,

$$P(\chi^2_{(L-1)(K-1)} > c) > P(\chi^2_{(L-2)(K-1)} > c)$$

$$p_c(S) > p_c(S')$$

Ces deux dernières propositions (9 et 10) sont extrêmement importantes car elles traduisent :

- la sensibilité du modèle à la taille de l'effectif étudié, le même arbre construit sur un échantillon de plus grande taille sera d'autant crédible;
- la sensibilité de la mesure à la complexité du modèle, regrouper des subdivisions ayant la même distribution de classes renforce le pouvoir prédictif du modèle.

5.4 Comparaisons

Nous n'avions pas voulu répondre au débat pour ou contre l'élagage dans la section précédente car il nous semblait peu approprié de comparer des méthodes répondant à des paradigmes différents : si la règle d'arrêt peut être fondée sur un test statistique, l'élagage lui peut reposer sur une optimisation de l'erreur estimés de différentes manières.

Il en est autrement dans ce chapitre, nous disposons d'une mesure de qualité globale, les deux stratégies sus-citées deviennent des alternatives pour trouver la partition optimale. Dans notre étude, nous avons utilisé la version gloutonne de construction d'arbre de [Wallace et Patrick, 1993] qui est une extension aux attributs multivalués du système d'encodage décrit dans la section 5.2.2.

Les résultats de cette expérimentation amènent différentes remarques :

- les différences en temps de calcul que nous avons notées sont énormes, l'adoption de la règle d'arrêt permet d'élaborer rapidement l'arbre, ce n'est pas le cas de l'élagage d'autant plus que nous avons fixé une taille limite des feuilles à un individu, ce qui les amène à rechercher la partition la plus fine avant de réduire l'arbre;

	Règle d'arrêt (1)	Elagage (2)	Ecart
autos	0.606 ± 0.054	0.639 ± 0.067	$(1) < (2)$ **
breast	0.928 ± 0.020	0.927 ± 0.020	
car	0.872 ± 0.011	0.872 ± 0.011	
machine	0.950 ± 0.031	0.948 ± 0.034	
credit	0.934 ± 0.038	0.934 ± 0.038	
flag	0.599 ± 0.108	0.586 ± 0.103	
hepatitis	0.812 ± 0.044	*n/a*	
ionosphere	0.896 ± 0.029	0.903 ± 0.024	
iris	0.942 ± 0.019	0.942 ± 0.019	
lung_cancer	0.470 ± 0.230	*n/a*	
pima	0.726 ± 0.023	0.725 ± 0.024	
vote	0.917 ± 0.074	0.917 ± 0.074	
wave	0.702 ± 0.041	0.696 ± 0.047	
wine	0.913 ± 0.054	$0.906 \pm .049$	
zoo	0.836 ± 0.072	0.815 ± 0.089	$(1) > (2)$*

TAB. 5.2 – *Comparaisons des performances entre l'arrêt de l'expansion et l'élagage dans les arbres*

- au final, les arbres construits sont le plus souvent identiques, la taille est parfois légèrement plus élevée dans le cas de l'élagage;
- le gain de l'élagage sur ces bases est nul comme on peut le constater dans le tableau 5.2.

Ces résultats montrent que dans la recherche d'une partition optimale en utilisant une mesure arbitrant entre complexité et précision, la stratégie de l'élagage alourdit inutilement les calculs et n'amène aucune amélioration en terme de performances et de concision du classifieur.

5.5 Conclusion

L'adoption de mesures arbitrant entre complexité et précision, permettant ainsi de juger de la qualité globale de la partition, constitue véritablement une avancée majeure dans l'élaboration des arbres de décision. Elle a permis une généralisation facile par les graphes et déplacé la problématique de l'inférence vers l'optimisation. Le succès de cette méthode repose bien entendu sur la corrélation présumée entre l'indicateur utilisé et le taux d'erreur en généralisation.

De ce point de vue, notre approche par l'analyse de variance sur tableau de contingence nous a permis de montrer que l'inférence par graphes sur données catégorielles pouvait être ramenée au schéma de régression classique, la divergence ne tient finalement qu'à la nature de l'opération de différence mise en oeuvre pour calculer les variances.

Sur un tout autre point, nous constatons que l'élagage n'apporte rien face à la règle d'arrêt dans ce contexte. Ce résultat doit cependant être tempéré, en effet les travaux de [Mingers, 1989a] semblent montrer une meilleure efficacité des méthodes mettant en oeuvre un échantillon supplémentaire pour l'élagage, mais cela introduit un coût supplémentaire que l'on n'est pas toujours disposé à supporter.

Chapitre 6

Extraction et traitement des bases de règles issues des graphes d'induction

6.1 Introduction

Une des finalités les plus avantageuses des classifieurs issus de l'apprentissage est de les utiliser dans les systèmes à base de connaissances [25]. Ces derniers, rappelons-le, se comportent comme des experts et proposent des solutions à un utilisateur humain qui doit prendre une décision. Pour convaincre son interlocuteur, le système doit pouvoir justifier sa réponse et expliquer comment il y est arrivé. A cette fin, il est absolument nécessaire que la connaissance manipulée soit alors intelligible à l'homme [Rubiello, 1997]. Sans entrer dans une polémique cognitiviste, il semble évident que les connaissances exprimées à l'aide de règles de production soient plus faciles à appréhender que les équation d'hyperplans morcelant l'espace de représentation des individus. D'ailleurs la plupart des systèmes experts actuels utilisent ce type de représentation [Mowforth, 1986].

Souvent les méthodes d'induction de règles à partir d'exemples cherchent le classifieur en optimisant directement l'expression d'une règle dans l'algorithme d'exploration des meilleures solutions [Michalski, 1983] [Clark et Boswell, 1991]. Il n'y a donc pas de passage à gérer entre la phase d'apprentissage et la phase d'exploitation. D'autres méthodes, en revanche, possèdent leur propre système de représentation dont elles essaient de trouver la meilleure expression, avant de la transformer dans un second temps en base de règles, moyennant quelques approximations et pertes d'informations que l'on essaie de minimiser. La traduction d'un réseau de neurones, qui souffrent énormément de leur opacité, en une base de règles a connu notamment d'importants développements ces dernières années [Andrews *et al.*, 1995] [Craven et Shavlik, 1994b]. Des travaux ont été également menés pour traduire un réseau de neurones en un arbre de décision

25. Système expert

[Craven et Schavlik, 1996].

Les graphes d'induction se démarquent par une souplesse qui font d'ailleurs leur popularité. D'une part, ils sont utilisables tels quels dans un processus de prise de décision, l'arborescence permet de visualiser la conclusion correspondant à un individu à classer; d'autre part, il en existe une traduction exacte sous la forme de base de règles sans perte d'informations. Dès lors, le classifieur issu de l'apprentissage peut être directement intégré dans un système expert, sans précautions particulières. Toutefois, il serait dommage de procéder ainsi. En effet, les bases de règles se prêtent à trois types d'opérations que l'on peut difficilement réaliser sur un graphe :

- la détection de "formes" : chaque noeud du graphe est une règle potentielle, nous pouvons nous assurer de la pertinence de sa conclusion avant de l'insérer dans le système expert [Rakotomalala *et al.*, 1996];

- la simplification : dans certaines configurations, les arbres peuvent buter sur les insuffisances de son système de représentation, on pense notamment aux expressions en forme normale disjonctive, le traitement de la base de règles permet de s'affranchir des problèmes inhérents, notamment la réplication des sous-arbres et les sommets à trop faibles effectifs [Dietterich, 1990];

- la fusion de bases de connaissances : qu'elles soient issues d'un autre processus d'induction ou d'un interview d'expert, les règles d'origines différentes peuvent être assemblées dans le même système. Il est même possible alors, moyennant des hypothèses restrictives sur le degré de certitude des règles de procéder à une simplification de la base ainsi construite [Rabaseda *et al.*, 1996a].

L'organisation de ce chapitre est la suivante : tout d'abord, nous définissons ce que nous entendons par "règles" et les mesures de pertinence associées. Nous y exposerons en particulier une procédure de validation statistique inspirée des travaux de [Gras et Lahrer, 1993] et [Lerman *et al.*, 1981]. Dans la section suivante, nous discuterons de la problématique d'assignation d'une conclusion à un noeud, nous y verrons notamment que l'affectation à la classe majoritaire obéit à une logique qu'il est nécessaire de préciser. La section qui suit est consacrée au coeur de notre chapitre : l'extraction des règles dans un graphe d'induction. Nous présenterons la procédure triviale, mais également les autres procédures où l'on essaie de tenir compte de la particularité des règles face aux graphes. Enfin, nous abordons la phase de traitement des bases de règles en présentant deux familles de méthodes de simplification de règles : la première symbolique, ne tenant pas compte des informations en provenance de l'échantillon d'apprentissage; la seconde purement numérique, fondée sur une optimisation d'une mesure que l'on précisera.

6.2 Caractérisation des règles

Dans notre travail, une règle s'écrit :

Si *Prémisse* **Alors** *Conclusion*

où

- *Prémisse* est une conjonction de propositions de type attribut-valeur. Par exemple pour la variable $Sexe$, une proposition possible serait

$$Sexe = Feminin$$

- *Conclusion* désigne la classe que l'on veut prédire. Si le problème est de savoir la présence ou l'absence de barbe chez un être humain, une conclusion éventuelle sera

$$Barbe = Absent$$

En reprenant notre exemple ci-dessus, la règle devient

$$\text{Si } (Sexe = Feminin) \textbf{ Alors } Barbe = Absent$$

6.2.1 Notations, formulations et position du problème

La partie prémisse d'une règle désigne un sous-ensemble $\Omega_{.R}$ de la population Ω. Par exemple, les personnes de sexe féminin représentent un peu plus de la moitié de la population française. Comme dans tout problème d'induction, nous travaillons en réalité sur un échantillon d'apprentissage Ω^a de la population, les différentes règles en décrivent une partition, pas nécessairement disjointe (graphique 6.1). En termes ensemblistes, une règle peut donc être décrite comme la circonscription d'un sous-ensemble de l'échantillon initial dans lequel il existe un mécanisme de désignation de la classe.

Nous noterons Π l'ensemble des prémisses conjonctives que l'on peut construire à partir des attributs $X_1, ..., X_p$ prenant respectivement leurs valeurs dans$\{x_{1,1}, \ldots, x_{1,\eta_1}\}, \ldots, \{x_{p,1}, \ldots, x_{p,\eta_p}\}$. Soit π_l, une prémisse particulière $\pi_l \in \Pi$,

$$\begin{aligned} \pi_l &: \Omega \longmapsto \{0,1\} \\ \pi_l(\omega) &= 1 \textit{ si l'individu vérifie la prémisse } \pi_l, \\ &= 0 \textit{ sinon} \end{aligned}$$

On notera n_{π_l} le nombre d'attributs qui interviennent dans la prémisse π_l. Une règle R_t sera définie comme la donnée d'une prémisse π_l et d'une conclusion y_k.

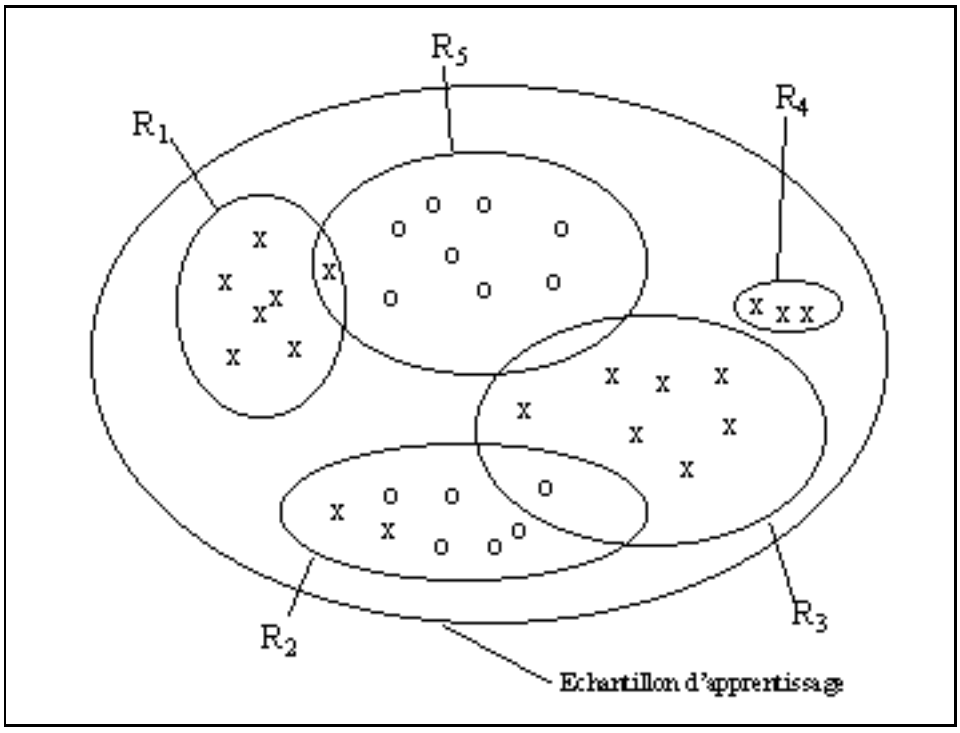

FIG. 6.1 – *Les règles R_1 à R_5 decrivent des sous-ensembles de l'échantillon d'apprentissage*

$$R_t = (\pi_l, y_k)$$

Nous définissons les sous-ensembles suivants :

- $\Omega^a_{k.} = \{\omega \in \Omega^a \,/\, Y(\omega) = y_k\}$, l'ensemble des individus de l'échantillon d'apprentissage portant l'étiquette y_k, avec $n_{k.} = Card(\Omega^a_{k.})$;
- $\Omega^a_{.l} = \{\omega \in \Omega^a \,/\, \pi_l(\omega) = 1\}$, l'ensemble des individus de l'échantillon d'apprentissage couverts par la prémisse π_l, avec $n_{.l} = Card(\Omega^a_{.l})$;
- $\Omega^a_{kl} = \{\omega \in \Omega^a \,/\, [\pi_l(\omega) = 1]\, et\, [Y(\omega) = y_k]\}$, l'ensemble des individus de l'échantillon d'apprentissage couverts par la prémisse π_l et portant l'étiquette y_k, $n_{kl} = Card(\Omega^a_{kl})$.

Afin de fixer les idées, nous présentons quelques exemples de règles dans un problème à deux classes mettant en jeu trois attributs booléens, avec les effectifs associés (Table 6.1).

Formellement, notre propos consiste à poser deux questions :

1. connaissant la prémisse π_l et les effectifs associés ($n_{.l}$, n_{kl}), quelle est la classe y_k que nous pouvons lui associer ?

$Y[n_{1.} = 10, n_{2.} = 10]$
R_1 : Si X_1 $Alors$ $Y[n_{11} = 8, n_{21} = 1]$
R_2 : Si X_1 et X_2 $Alors$ $Y[n_{12} = 5, n_{22} = 0]$
R_3 : Si X_1 et X_2 et X_3 $Alors$ $Y[n_{13} = 3, n_{23} = 0]$

TAB. 6.1 – *Trois règles construites sur un échantillon d'apprentissage contenant 10 individus de la classe y_1 et 10 individus de la classe y_2*

2. quel crédit doit-on accorder à la règle ainsi construite?

Nous nous concentrons sur la deuxième question dans cette section, nous répondrons de manière approfondie à la première dans la section suivante.

6.2.2 Les mesures d'évaluation des règles

Critères d'évaluation

Chaque règle $R_t = (\pi_i, y_k)$ peut être évaluée à partir de l'échantillon d'apprentissage. Du point de vue de la reconnaissance de formes, nous essayons de savoir si l'on a réussi à extraire une forme significative i.e représentant un mécanisme de désignation de la classe à partir de la prémisse dans la population globale Ω.

Pour juger de la crédibilité des règles à partir du fichier d'apprentissage, [Goodman et Smyth, 1988] ont proposé deux exigences antinomiques :

- d'une part, si la prémisse de la règle est peu restrictive, elle aura tendance à couvrir un grand nombre d'individus pour lesquels on pourra associer une conclusion, c'est le critère de **généralité**. Ce faisant, nous nous exposons certainement à fort taux d'erreur car la description des individus est trop pauvre. Au total, la base de règles pourra classer tous les individus mais avec un taux d'erreur peu satisfaisant.

- d'autre part, si la prémisse de la règle est très spécialisée, couvrant une fraction très faible de la population, elle présentera certainement un très faible taux d'erreur en resubstitution, c'est le critère de **précision**. Cette forte spécialisation n'est pas une garantie de performances en généralisation car la trop grande spécificité des règles peut signifier tout simplement une ingestion de bruits éxagérée qui l'empêche de reconnaître un individu n'ayant pas servi lors de la phase d'apprentissage.

Le critère d'évaluation des règles doit donc trouver la juste mesure entre ces deux exigences. Dans le cas des graphes d'induction, le critère peut être appliqué a posteriori, lorsque la règle est construite, pour juger de sa pertinence; dans d'autres cas, il peut être utilisé directement pour induire les meilleures règles dans un processus d'apprentissage [Smyth et Goodman, 1992].

Familles de mesures

Compte tenu de la nature de l'information prise en compte dans les différentes mesures utilisées, nous pouvons définir trois familles de mesures d'évaluation des règles :

- *les fonctions d'évaluation locales* : les fonctions qui entrent dans cette dénomination ne prennent en compte que l'information apportée par le sous-ensemble décrit par la prémisse $(\Omega^a_{kl}, k \cup \Omega^a_{kl})$. La première, issue de la théorie de l'estimation statistique, est tout simplement l'estimateur du maximum de vraisemblance de la probabilité d'apparition d'une classe; la seconde, issus des travaux sur la théorie de l'information est une mesure de la quantité de bits nécessaires pour désigner une classe dans un sous-ensemble donné. Ces mesures répondent uniquement au critère de précision, on leur reproche souvent d'être par trop optimistes.

- *les fonctions d'évaluations locales à correction ad-hoc* : pour pallier les insuffisances des indicateurs précédents, des variantes ont été proposées, notamment pour tenir compte de la taille de l'échantillon dans l'estimation de l'indicateur. On essaie ainsi d'introduire le critère de généralité. Nous y recensons la borne basse de l'intervalle de confiance du taux de succès de classement par resubstitution, et l'estimateur laplacien du taux de succès.

- *les fonctions d'évaluations tenant compte de la complexité de description* : les mesures qui entrent dans cette famille essaient de trouver le meilleur compromis entre le coût de description de la règle et des exceptions à cette règle. Le critère de généralité est introduit par la pénalisation des règles trop complexes, qui nécessairement décrivent une fraction trop faible de la population; le critère de précision est inclus grâce à la description des exceptions qui peut être très coûteuse si la règle n'est pas assez précise.

- *les fonctions d'évaluation tenant compte de la distribution initiale des classes* : nous devons garder à l'esprit que la règle décrit une fraction des individus de l'échantillon d'apprentissage. Pour appréhender la qualité de la règle, il faut non seulement mesurer le différentiel de distribution des classes dans les différents sous-ensembles $(\Omega^a_{kl}, \Omega^a_{k.})$, mais également vérifier leurs cardinalités relatives $(n_{.l}, n)$. La j-mesure de [Goodman et Smyth, 1988] en fait partie, nous verrons également dans la sous-section suivante que l'adoption d'une formulation statistique permet de définir une mesure d'implication qui possède les même propriétés.

Dans ce qui suit, nous décrirons ces mesures en détaillant leurs particularités.

Estimateur statistique de la probabilité d'apparition d'une classe : la fréquence $F(R_t)$

L'idée ici consiste à dire : "une règle est d'autant meilleure qu'elle a une faible probabilité de se tromper, donc une forte probabilité de classer convenablement; dès lors, pour qualifier une

règle, il suffit d'estimer cette probabilité". L'estimateur de la probabilité qui semble s'imposer est l'estimateur du maximum de vraisemblance, qui est sans biais. Il s'agit de la fréquence. La fonction d'évaluation d'une règle est donc $F(R_t) = \frac{n_{kl}}{n_{.l}}$. Dans notre exemple $F(R_3) = \frac{3}{3} = 1$ contre $F(R_1) = \frac{8}{9} = 0.88$, ce qui nous fait choisir la troisième règle.

Le principal reproche adressé à cet indicateur est l'absence d'information sur la variabilité de l'estimateur ponctuel de cette probabilité. Il est évident qu'il sera d'autant plus fiable que le nombre d'individu associé à la règle est élevé, ici nous ne pourrons pas distinguer R_2 de R_3. De plus, il a été prouvé que cet estimateur est trop optimiste dans la mesure où justement la méthode a essayé de l'optimiser pour choisir la meilleure règle.

Nombre de bits nécessaires pour désigner une classe : l'entropie de Shannon $H(R_t)$

Utilisée par exemple dans le premier algorithme CN2 [Clark et Niblett, 1989], elle mesure la quantité de bits nécessaires pour connaître la classe d'un individu dans un sous-ensemble d'observations [Shannon et Weaver, 1949]. L'expression associée est :

$$H(R_t) = -\sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}} \log_2(\frac{n_{kl}}{n_{.l}})$$

Selon [Clark et Niblett, 1989], elle présente l'avantage de favoriser le choix de sous-ensembles de grande taille avec peu d'individus hors classes de focalisation. Ainsi, dans un problème à 4 modalités, le choix en faveur de $Pr(0.7, 0.3, 0, 0)$ contre Pr$(0.7, 0.1, 0.1, 0.1)$ permet de dépasser les insuffisances de l'indicateur précédent (estimation du maximum de vraisemblance du taux de succès). Malgré ces qualités, nous remarquerons que cet indicateur fût abandonné au profit de l'estimateur laplacien du taux d'erreur dans la deuxième version de CN2 [Clark et Boswell, 1991], notamment à cause de sa tendance au sur-apprentissage.

Borne basse de l'intervalle de confiance du taux de succès de classement $B(R_t)$

Sachant que l'estimation ponctuelle de la probabilité à l'aide de la fréquence est trop souvent optimiste, et que ce biais est vraisemblablement une fonction décroissante de l'effectif associé à la règle, différents auteurs [Kalkanis, 1993] ont proposé d'utiliser la borne basse de l'intervalle de confiance de cette fréquence : moins il y aura d'individus couverts par la règle, plus elle sera éloignée de l'estimation ponctuelle. Ainsi, la fonction d'évaluation proposée tient compte de la variabilité de la mesure proposée.

Le principe de calcul est le suivant : chaque individu désigné par une règle peut être bien ou mal classé, les n_{kl} individus sur les n_l, correspondent donc à la répétition de l'événement "classement adéquat", on peut avancer que la variable aléatoire $N_{kl} \rightsquigarrow \beta(n_{.l}, p)$ suit une loi binomiale, avec p la probabilité de bien classer à estimer. La fonction d'évaluation, que [Quinlan, 1990]

qualifie d'estimateur pessimiste du taux de succès, est la quantité $B(R_t)$ qui est solution de l'équation

$$\sum_{i=0}^{n_{kl}} n_{kl} i (B(R_t))^i (1 - B(R_t))^{n_{kl}-i} = \alpha$$

Sous certaines hypothèses ($n_{kl} \geq 5$), nous pouvons utiliser l'approximation normale. Dès lors, $B(R_t) = F - u_\alpha * \sqrt{\frac{F(1-F)}{N_\Pi}}$, où u_α est la valeur critique de la loi normale pour un niveau de confiance α.

Le choix du risque critique α revient à l'utilisateur et peut être optimisé en utilisant différentes heuristiques. En l'absence d'informations supplémentaires, nous fixons $\alpha = 0.25$ [Quinlan, 1993a].

Estimateur laplacien du taux de succès $L(R_t)$

Cet estimateur part de la théorie de l'information, notamment en abordant le problème du point de vue du codage. Chaque individu couvert par la règle peut être considéré comme un évènement dans l'espace $\{y_1, y_2\}$, un codage incrémental consiste à encoder l'évènement $(Ev_i = y_1)$ (resp. $Ev_i = y_2$) avec un code de longueur $-\log[\Pr(Ev_i = y_1)]$ (resp. $-\log[\Pr(Ev_i = y_2)]$).

Lorsque Ev_{i-1} évènements ont été transmis, le codage incrémental associé à l'évènement Ev_i met à jour ces probabilités de la manière suivante :

$$\Pr(Ev_i = y_1) = f(Ev_1, ..., Ev_{i-1}).$$

Dès lors, l'expression ci-dessus dépend de la probabilité initiale $\Pr(Ev_i = y_1)$. Si nous choisissons une distribution Beta [26] symétrique de la forme $f(p) = \frac{p^\alpha (1-p)^{\alpha-1}}{\beta(\alpha,\alpha)}$. La probabilité d'obtenir un individu supplémentaire de classe y_1 s'écrit [Wallace et Boulton, 1968] alors :

$$\Pr(Ev_i = y_1) = \frac{n_{1l} + \alpha}{n_{.l} + 2 * \alpha}$$

Dans un problème à plus de deux classes, la généralisation de cet indicateur est assez naturelle

$$L(R_t) = \frac{n_{kl} + \alpha}{n_{.l} + K\alpha}$$

Cet estimateur, dit laplacien, du taux de succès peut donc s'interpréter comme la probabilité d'obtenir un individu supplémentaire de la classe y_1 après avoir observé la distribution empirique des classes sur $n_{.l}$ individus. Nous supposons que ces classes sont a priori équiprobables sur le premier individu (évènement) observé, dans ce cas $\alpha = 1$. On peut conjecturer que cette statistique est autrement plus réaliste que l'estimateur par resubstitution (maximum de vraisemblance) car elle propose une évaluation de la probabilité de succès sur un individu de l'échantillon test que

26. Cette distribution est souvent associée à des estimations de probabilités fournis par des experts [Aïvazian 1986]

l'on aura à classer. Nous remarquons que cet estimateur est le même que l'estimateur bayésien des probabilités que nous avons développé dans le chapitre sur les mesures de qualité des partitions, d'autres auteurs [Clark et Boswell, 1991] en attribuent la paternité à [Cestnik, 1990].

Fonction d'évaluation tenant compte de la complexité de la description $C(R_t)$

Issue à l'origine des travaux de [Rissanen, 1978], l'introduction de la complexité de la description en reconnaissance de formes - la théorie de la description minimale des messages - fût en grande partie l'oeuvre de [Quinlan et Rivest, 1989] qui impulsèrent un nouveau champ de recherche très prolifique ces dernières années; elle eût également pour mérite de réveiller les ardeurs de chercheurs qui ont travaillé sur des domaines connexes depuis plusieurs années, notamment sur la théorie du message de longueur minimum [Georgeff et Wallace, 1984]. La base théorique est dans l'assertion selon laquelle la règle qui minimise sa longueur de description est la règle qui maximise la probabilité $\Pr(\mathit{Règle}/\mathit{DonnéesObservées})$ de l'obtenir sachant les données [Wallace et Freeman, 1987].

D'une certaine manière, les deux (MDL, MML) stratégies se rejoignent. Considérons deux individus : un émetteur et un receveur. Chacun des deux possède la liste des individus observés et la description de leurs attributs. Le problème consiste à trouver le message de longueur minimale qui permettra de transmettre la description des classes des individus observés de l'émetteur au récepteur. Une méthode simple serait d'énumérer la classe d'appartenance de chaque individu, mais elle peut être coûteuse du point de vue du nombre de bits nécessaire pour transmettre ce message. Il peut être profitable de trouver une théorie - ici, la règle de production - qui résume au mieux la relation entre les attributs et la classe, et qui ne soit pas trop coûteuse à décrire. La différence conceptuelle entre les deux stratégies (MDL et MML) réside dans le fait que la première revient à comparer la description complète de chaque observation avec la description de la théorie additionnée à la description des exceptions à cette théorie, alors que la seconde la compare à la description de la théorie additionnée à la description des données connaissant celle-ci. Il n'en reste pas moins que la quantité calculée est quasiment la même dans les deux cas.

Dans tout ce qui suit, nous considérerons que tous les attributs continus auront été discrétisés globalement et que les informations sur les points de discrétisation ont été préalablement transmises, donc nous ne nous préoccuperons que de l'encodage des règles contenant des attributs qualitatifs.

Encodage d'une règle avec le principe du message de description minimale L'encodage d'un arbre de décision a fait l'objet d'un débat passionné ces dernières années [Wallace et Patrick, 1993]. Il semble par contre que la situation soit différente en ce qui concerne la description d'une règle de production, la méthode proposée par [Quinlan, 1993a] semble suffisamment stable pour que nous l'adoptions ici.

La partie prémisse d'une règle est formée d'une conjonction de propositions. Pour encoder la règle, nous devons donc tour à tour les spécifier, sauf que l'ordre d'envoi importe peu. Dans ce cas, Quinlan propose de réduire la description de la théorie par la quantité $\log_2(Nombre\_de\_propositions!)$.

Prenons un exemple simple pour illustrer nos propos. Soient deux attributs {A,B}, prenant respectivement leurs valeurs dans {A1,A2,A3} et {B1,B2}, candidats pour décrire une règle. Si celle-ci s'écrit *Si (A=A1) et (B=B2) Alors...,* la longueur du code associé sera $[(\log_2(2) + \log_2(3)) + (\log_2(1) + \log_2(2)) - \log_2(2!)] = \log_2(5)$. Cette équation se lit : il y a deux attributs candidats, il faut 1 bit pour savoir qu'il s'agit de A, et $\log_2(3)$ bits pour savoir que c'est la modalité A1 qui est mis à contribution; on sait donc que l'attributs suivant sera B (on a besoin de $\log_2(1) = 0$ bits pour le savoir), et on a besoin de $\log_2(2)$ bits pour savoir que la proposition est formée par la modalité B2. Enfin, sachant que notre message peut être envoyé dans le sens [(A=A1) et (B=B2)] ou [(B=B2) et (A=A1)], ce coût [Quinlan, 1993a] sera réduit de $\log_2(2!) = 1$ bit.

Le coût de description d'une règle s'écrit donc

$$C_{Theorie}(R_t) = \sum_{j=0}^{n_{\pi_l}} (\log(p-j) + \log(\eta_j)) - \log(n_{\pi_l}!)$$

Encodage des exceptions Connaissant maintenant la règle, nous devons calculer la longueur du code nécessaire pour décrire les exceptions à cette règle. A cette fin, [Quinlan, 1994] définit deux indicateurs particuliers:

- les "faux positifs" correspondent aux individus couverts par la règle, et n'appartenant pas à la classe de conclusion y_k, $fp = n_{.l} - n_{kl}$;
- les "faux négatifs" sont les individus issus de l'ensemble d'apprentissage, non-couverts par la règle, et pourtant appartenant à la classe de conclusion y_k, $fn = n_{k.} - n_{kl}$.

Pour encoder les exceptions, nous devons donc spécifier : d'une part, quels sont les individus couverts par la règle, et quels en sont les faux positifs; d'autre part, quels sont les individus non-couverts par la règle, et lesquels sont des faux négatifs. Le coût d'encodage des exceptions se calcule à l'aide de la formule suivante :

$$\begin{aligned} C_{Exceptions}(R_t) \quad = \quad & \log(n_{.l}+1) + fp * (-\log(\frac{fp}{n_{.l}})) + (n_{.l} - fp) * (-\log(1 - \frac{fp}{n_{.l}})) \\ & + \log(n - n_{.l} + 1) + fn * (-\log(\frac{fn}{n - n_{.l}})) + (n - n_{.l} - fn) * (-\log(1 - \frac{fn}{n - n_{.l}})) \end{aligned}$$

Au total, le coût de description d'une règle s'obtient par simple addition des deux quantités ci-dessus

$$C(R_t) = C_{Theorie}(R_t) + C_{Exceptions}(R_t)$$

Une interprétation aisée et une forte cohérence théorique sont certainement parmi les aspects les plus séduisants de cette approche. Cependant, notre objectif étant de classifier avec un minimum d'erreur, [Quinlan, 1995] pense qu'en réintroduisant explicitement le taux d'erreur en resubstitution dans le calcul des exceptions, nous pourrions obtenir de meilleures performances. Nous ne testerons cependant pas cette variante car, à l'inverse de la méthode originelle, la justification de ce biais est purement empirique.

La j-Measure $J(R_t)$

Chaque règle apporte de l'information. [Goodman et Smyth, 1988] ont proposé une mesure qu'ils interprètent comme la diminution moyenne de bits nécessaires pour désigner une classe entre la distribution a priori $P(Y = y_k)$, et la distribution a posteriori $P(y = y_k/\pi_l)$.Cette quantité est estimée par :

$$j(y_k, \pi_l) = \frac{n_{kl}}{n_{.l}} \log(\frac{\frac{n_{kl}}{n_{.l}}}{\frac{n_{k.}}{n}}) - \frac{n_{.l} - n_{kl}}{n_{.l}} \log(\frac{\frac{n_{.l}-n_{kl}}{n_{.l}}}{\frac{n-n_{k.}}{n}})$$

elle représente la capacité à prédire la classe à l'aide de la prémisse. Sachant que la règle est destinée à la prédiction, il faut qu'elle ait une probabilité raisonnable d'apparaître, pour cette raison, les auteurs proposent finalement l'indicateur :

$$J(y_k, \pi_l) = P(\pi_l) * j(y_k, \pi_l)$$

où $P(\pi_l)$ est estimé par $\frac{n_{.l}}{n}$. Au total, une règle avec une forte valeur de $J(.)$ aura de bonnes capacités prédictives et une probabilité convenable d'apparaître.

Le principal intérêt de cette mesure est qu'elle tient compte d'une part de la taille relative des sous-ensembles, d'autre part du différentiel de distribution des classes. Toute l'information disponible dans les différents sous-ensembles $(\Omega^a, \Omega^a_{k.}, \Omega^a_{kl}, \Omega^a_{.l})$ est donc utilisée, à la différence des mesures locales (entropie, fréquence...) qui ne mettent en oeuvre que des informations parcellaires $(\Omega^a_{kl}, \Omega^a_{.l})$.

6.2.3 La validation statistique d'une règle : l'intensité d'implication

Toutes les mesures ci-dessus, à l'exception de celle fondée sur la description minimale des messages, permettent d'établir une hiérarchie parmi les règles, mais ne nous informe pas sur leur pertinence. Certes, il serait possible de fixer une valeur limite pour décider qu'une règle est intéressante ou non. Dans [Zighed et Rakotomalala, 1996a] par exemple, on propose de fixer un seuil de spécialisation (une valeur limite du taux de succès associée à la règle) et un effectif minimum pour valider une règle. Ces stratégies, même si elles relèvent du bon sens et respectent

à la lettre les critères de généralité-précision, ne possèdent pas de fondements mathématiques véritables, et en tant que telles reposent sur la perspicacité de l'expert.

La situation est différente en ce qui concerne le coût de description minimale. On peut confronter le coût de description de tous les individus avec le coût de description de la règle additionnée du coût de description des exceptions : on décide de la rejeter si le deuxième terme est supérieur au premier. Si cette procédure est sans conteste valable pour décider de la pertinence d'une règle, il lui manque la capacité à discerner différentes hypothèses restrictives quant au choix d'adoption d'une décision comme il est d'usage en théorie des tests. Dans ce dernier cas, nous pouvons aboutir à des résultats différents en choisissant des risques critiques répondant à des exigences de précision différentes. Dans l'optique d'une exploration automatique des données, nous pouvons fixer des valeurs standards (1%, 5%, 10%) de risque de première espèce, qui est la probabilité d'accepter l'existence d'un phénomène de causalité alors qu'en réalité la prémisse et la classe sont indépendantes.

Test fondé sur la statistique du contre-exemple

Les premiers travaux en la matière ont été l'oeuvre de [Gras, 1979] qui a étudié la causalité entre deux attributs qualitatifs binaires dans la recherche d'enchaînements successifs de comportements. Le test repose essentiellement sur la statistique du contre-exemple, qui dans le cadre de notre travail, correspond au nombre d'individus n'appartenant pas à la classe de conclusion y_i dans le sous-ensemble décrit par la prémisse.

Formellement, le test s'énonce de la manière suivante : "connaissant la distribution des individus dans l'échantillon de départ $P(Y = y_k)$, estimé par $\frac{n_k}{n}$, on décide que la prémisse π_l mène à la conclusion $(Y = y_i)$ si et seulement si la quantité $\sum_{k \neq l} n_{kl}$ est *invraisemblablement petite* [Gras et Lahrer, 1993]. La statistique $CE(R_t) = \sum_{k \neq l} n_{kl}$ permet de mesurer l'intensité d'implication entre deux attributs.

L'étude de la distribution de $CE(R_t)$ a été effectuée dans [Lerman *et al.*, 1981], selon différentes hypothèses de tirage elle peut suivre une loi hypergéométrique, binomiale ou de Poisson. C'est cette dernière qui nous intéresse, en effet elle est non symétrique [Gras et Ratsimba-Rajohn, 1996], l'étude de la rareté des "contre-exemples" n'est pas simplement le dual de l'étude de l'abondance des "exemples".

La formulation du test est la suivante :

$$H_0 \ : \ \pi_l \ et \ y_i \ \ sont \ ind\acute{e}pendants$$
$$H_1 \ : \ \pi_l \Rightarrow y_i$$

Sous l'hypothèse nulle H_0, [Lerman *et al.*, 1981] a montré que $CE(R_t)$ suivait une loi de

Poisson de paramètre

$$\lambda = \frac{n_{.l} \times (n - n_{i.})}{n}$$

La région critique de rejet de l'hypothèse nulle du test devient ainsi

$$CE(R_t) \leq p\alpha(\lambda)$$

où α est le risque de première espèce associé au test, et $p\alpha(\lambda)$ est la valeur critique lue dans la table de la loi de Poisson de paramètre λ au point de pourcentage α.

L'interprétation du test est relativement aisée : moins il y a de contre-exemples à y_i dans le sous-échantillon circonscrit par la prémisse π_l, plus on pourra conclure à la présence significative de cette classe.

Intensité d'implication

Comme tous les tests, la région critique peut être réécrite en

$$\Pr(Poisson(\lambda) \leq CE(R_t)) \leq \alpha \tag{6.1}$$

On peut considérer qu'une règle est valide, c'est à dire s'éloignant significativement de la situation aléatoire en faveur de l'absence de contre-exemples à la classe y_i, si elle vérifie l'équation 6.1. En fixant différentes valeurs de α, nous pouvons choisir notre niveau d'exigence de précision des règles.

Cette deuxième formulation est d'autant plus intéressante qu'elle nous permet de comparer directement des règles, et donc de les hiérarchiser dans une base de connaissances. L'expression que l'on nomme **Intensité d'Implication** se présente de la manière suivante :

$$I(R_t) = 1 - \Pr(Poisson(\lambda) \leq CE(R_t))$$

Elle varie entre 0 (indépendance totale) et 1 (causalité parfaite, implication logique) [Gras *et al.*, 1996], et répond bien aux exigences des critères généralité-précision tel qu'on peut le voir dans l'exemple du graphique 6.2.

Extension : détermination de l'effectif minimal associé à une règle

Comme nous l'avions dit plus haut, une règle servira à inférer sur de nouveaux individus que l'on voudra classer. Pour cela, il faut qu'elle ait un certain pouvoir prédictif que l'on évalue à l'aide du nombre d'individus qui lui est associé dans l'échantillon d'apprentissage. Ainsi, la plupart des méthodes d'induction par graphes permettent aux utilisateurs de fixer une taille minimale des sommets en deçà de laquelle la partition est refusée. D'ailleurs, dans [Zighed et Rakotomalala, 1996a] par exemple, une règle est invalidée si elle est portée par trop peu d'individus. Le problème est qu'il n'y a pas de manière explicite et automatique pour fixer

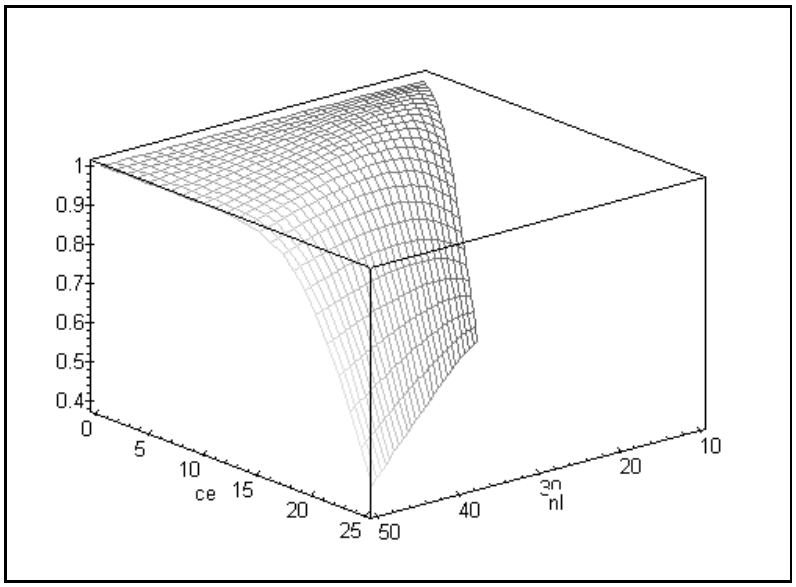

FIG. 6.2 – *L'intensité d'implication augmente à mesure que l'effectif associé à la règle n_l augmente et que le nombre de contre-exemples diminue (ce)*

cette taille limite. Certains auteurs [Quinlan, 1993a] fixent arbitrairement cette valeur à 2 sans véritable justification.

En utilisant l'intensité d'implication, cette contrainte est introduite implicitement, et la valeur limite peut être déterminée.

Definition 1 *Nous désignons par taille minimale d'une règle, la borne inférieure du nombre d'individus couverts par la règle tel qu'en l'absence de contre-exemples, elle soit validée par le test de Poisson.*

Nous avions écrit plus haut la région d'acceptation de la causalité 6.1, en explicitant la fonction de distribution de la loi de Poisson, elle devient

$$\sum_{m=0}^{CE(R_t)} \frac{(\frac{n_{.l}(n-n_{k.})}{n})^m}{m!} e^{(-\frac{n_{.l}(n-n_{k.})}{n})} \leq \alpha$$

Lorsque le nombre de contre-exemple $CE(R_t)$ est nul,

$$\begin{aligned} -\frac{n_{.l}(n-n_{k.})}{n} &\leq \ln(\alpha) \\ -n_{.l} &\leq \frac{n \times \ln(\alpha)}{(n-n_{k.})} \\ n_{.l} &\geq -\frac{n \times \ln(\alpha)}{(n-n_{k.})} \end{aligned}$$

Règle	Fréquence	Entropie	Upper CF	Laplacien	MDL	J-Meas.	I.Impl.
R_1	3	3	2	2	2	2	2
R_2	1	1	1	1	1	1	1
R_3	1	1	3	3	3	3	3

TAB. 6.2 – *Classement des règles*

Pour s'assurer qu'une règle soit valide, quelle que soit la conclusion, en l'absence de contre-exemple, il faut que l'effectif associé soit supérieur à $n^*_{.l}$. En deçà de cette valeur, la règle sera invalidée à coup sûr par le test.

$$n^*_{.l} = k = 1 \ldots KMax - \frac{n \times \ln(\alpha)}{(n - n_{k.})}$$

On remarquera que puisque $\alpha < 1$, $n^*_{.l}$ sera toujours supérieur à 0.

6.2.4 Evaluation empirique des mesures de qualité de règles

Comportement des mesures sur notre exemple

Prenons nos règles exemples ci-dessus pour observer le comportement des fonctions d'évaluations présentées. Intuitivement, il semble que la règle R_2 soit la meilleure, par contre la distinction entre R_1 et R_3 est moins tranchée, si R_1 répond mieux au critère de généralité (9 individus), R_3 l'emporte sur le critère de précision (0 erreur). Chaque règle sera classée selon la mesure utilisée (Table 6.2).

Conformément aux critiques qui ont été formulées précédemment, l'estimateur du maximum de vraisemblance du taux de succès et l'entropie ont une forte propension au sur-apprentissage, rien ne distingue la règle R_2 de R_3 alors qu'elle est manifestement meilleure (à précision égale, généralité plus grande). Il en est tout autrement des autres mesures qui placent bien R_2 devant R_3. En ce qui concerne R_1 et R_3, toutes les mesures prenant en compte la taille de l'effectif convergent, le gros écart de généralité (9 individus contre 3) l'emporte sur l'écart de précision (0.88 contre 1.0).

Comportement des mesures en prédiction

L'influence de l'appréciation de la règle est manifeste sur la règle extraite par un algorithme. Dans [Rakotomalala *et al.*, 1997a], nous avons montré qu'en utilisant l'algorithme CN2 [Clark et Boswell, 1991], les mesures tenant compte de l'information globale, soit sous la forme de la complexité de description, soit sous la forme de comparaison entre distributions conditionnelles et inconditionnelles des classes (MDL, j-Mesure, Intensité d'Implication), permettaient d'extraire les règles les plus générales. Du coup, à précision égale, la base construite était nettement moins complexe.

La problématique est similaire dans les graphes d'induction. Les noeuds du graphe sont tous des règles potentielles, il nous importe de les qualifier sur l'échantillon d'apprentissage. Le principal souci étant certes d'inférer au mieux sur le pouvoir prédictif, quantifié par le taux d'erreur en généralisation, mais également de déterminer les règles les plus générales i.e qui couvrent un nombre suffisant d'individus pour être généralisable dans la population..

6.3 Assignation d'une conclusion à un noeud du graphe

Dans la plupart des publications relatives aux graphes d'induction, l'assignation d'une conclusion à une feuille revient tout simplement à choisir la classe majoritaire [Quinlan, 1987a]. Lorsque deux classes ou plus possèdent le même effectif maximal, on peut décider de ne pas conclure, laissant ainsi la règle de conclusion indéterminée [Zighed *et al.*, 1992], ou tout simplement sélectionner aléatoirement l'une d'entre elles [Cestnik *et al.*, 1987b].

En réalité l'assignation d'une conclusion à la règle de la majorité correspond à un schéma de prise de décision très précis que nous expliciterons plus bas, nous verrons notamment que ses conditions d'utilisation sont assez restrictives selon le schéma d'échantillonnage utilisé.

6.3.1 La stratégie bayésienne : minimisation des coûts de mauvaise assignation

Soit π_l la prémisse associée à la règle correspondant au $l^{ème}$ noeud du graphe. Nous devons désigner une classe y_k qui sera la conclusion à la règle formée. Une manière très simple de justifier la décision est de choisir la classe qui minimise l'espérance de coût de mauvais classement bayésien.

Soit $c(k'/k)$ le coût associé à la sélection de la conclusion classe $y_{k'}$ alors que y_k est la bonne décision. L'espérance du coût de la décision s'écrit :

$$\sum_k c(k'/k) \times P(Y = y_k/\pi_l)$$

Nous devons donc choisir k' tel que

$$k' = 1, \ldots, K \min \sum_k c(k'/k) \times P(Y = y_k/\pi_l)$$

Le principal problème est alors d'estimer $P(Y = y_k/\pi_l)$ à partir de l'échantillon d'apprentissage. Soient :

- $p_{k.} = P(Y = y_k)$ la probabilité a priori qu'un individu porte l'étiquette y_k;
- $p_{.l} = P(\omega \in \Omega_{.l})$ la probabilité qu'un individu soit couvert par la prémisse π_l;

- $p_l(k) = P(\omega \in \Omega_{.l} / Y = y_k)$ la probabilité qu'un individu soit couvert par la prémisse π_l sachant qu'il porte l'étiquette y_k;
- $p_{kl} = P(Y = y_k/\pi_l)$ la probabilité a posteriori qu'un individu porte l'étiquette y_k sachant qu'il est couvert par la prémisse π_l.

La formule de Bayes nous permet d'obtenir :

$$\begin{aligned} p_{kl} &= \frac{p_{k.} \times p_l(k)}{p_{.l}} \\ p_{kl} &= \frac{p_{k.} \times p_l(k)}{k \sum p_{k.} \times p_l(k)} \end{aligned}$$

La problématique de la construction de classifieurs repose justement sur l'estimation du terme $p_l(k)$ [Celeux et Mkhadri, 1994]. Dans le cadre des graphes d'induction, nous pouvons prendre comme approximation [Gueguen, 1994][Pao, 1989]

$$\widehat{p}_l(k) = \frac{n_{kl}}{n_{k.}}$$

et l'estimation de p_{kl} devient

$$\widehat{p}_{kl} = \frac{p_{k.} \times \frac{n_{kl}}{n_{k.}}}{k \sum p_{k.} \times \frac{n_{kl}}{n_{k.}}}$$

Notre estimation dépend donc du choix de l'estimation de p_{kl} que l'on fait. Dans *l'hypothèse d'un tirage aléatoire simple dans la population* Ω, nous pouvons utiliser l'estimateur du maximum de vraisemblance

$$\widehat{p}_{k.} = \frac{n_{k.}}{n}$$

L'estimation $\widehat{p}_{kl}$ s'écrit alors

$$\begin{aligned} \widehat{p}_{kl} &= \frac{\frac{n_{k.}}{n} \times \frac{n_{kl}}{n_{k.}}}{k \sum \frac{n_{k.}}{n} \times \frac{n_{kl}}{n_{k.}}} \\ \widehat{p}_{kl} &= \frac{n_{kl}}{n_{.l}} \end{aligned}$$

Le choix de k' se fait en minimisant l'équation

$$k' \min \sum_k c(k'/k) \times \frac{n_{kl}}{n_{.l}} \tag{6.2}$$

Dans l'hypothèse de matrice de coûts symétriques unitaire de la forme

$$c(k'/k) = \begin{cases} 0 \;\; si \;\; k' = k \\ 1 \;\; si \;\; k' \neq k \end{cases}$$

Minimiser l'équation 6.2 revient à maximiser

$$\max_{k'} \frac{n_{k'l}}{n_{.l}} \tag{6.3}$$

Nous retrouvons bien la pratique courante de sélectionner la classe la plus fréquente dans le sommet correspondant à la prémisse π_l. Mais ce résultat est basé sur deux éléments fondamentaux qui sont souvent implicites dans les études :

- *la symétrie de la matrice des coûts* : si dans la plupart des cas, elle est licite. Il est des problèmes où elle n'est plus du tout appropriée. Si nous considérons par exemple le problème de la détection automatique du cancer du sein, il est certainement plus préjudiciable de conclure à tort l'absence de cancer et de laisser partir le patient, plutôt que de conclure à tort la malignité de la tumeur et de demander des examens supplémentaires;

- *le schéma de tirage de l'échantillon* : si le tirage aléatoire simple est un procédé d'école qui permet notamment l'application de nombreux résultats de statistique, il en est autrement dans les bases réelles. Toujours en médecine, il est d'usage d'effectuer ce que l'on appelle un tirage rétrospectif à K échantillons. Pour chaque classe y_k, on constitue par tirage aléatoire un échantillon de taille $n_{k.}$. Cette quantité $n_{k.}$ n'est plus aléatoire, on peut décider par exemple que tous les $n_{k.}$ sont égaux [Rabaseda, 1996].

Lorsque ces deux conditions ne sont pas remplies, la règle de décision d'affectation à la classe la plus fréquente est sujette à caution si l'on veut adopter le schéma bayésien.

6.3.2 Affectation par maximisation de l'intensité d'implication

Il y a un autre paradigme que l'on pourrait opposer à cette minimisation de la perte : la décision correspondant à la maximisation de l'intensité d'implication. En effet, notre objectif est de trouver des classes qui se démarquent significativement dans le sous-ensemble circonscrit par la prémisse. Il est alors naturel de choisir la classe possédant le moins de contre-exemples compte tenu de leur distribution initiale dans l'échantillon d'apprentissage. La règle de décision est donc de choisir la classe $y_{k'}$ telle que

$$\max_{k'=1,\ldots,K} I(R_t) \tag{6.4}$$

A la différence de la procédure précédente, elle n'introduit nulle part des hypothèses sur le mode de tirage de l'échantillon. Seuls, le différentiel de distribution, et les tailles respectives de l'ensemble initial et du sous-ensemble décrit par la prémisse nous permettent de choisir la décision la plus appropriée.

6.3.3 Conclusion, Non-Conclusions et règles à conclusion indéterminées

Règle à conclusion indéterminée

La question que l'on peut poser est : "est-il toujours nécessaire de conclure ?". Il est beaucoup de domaines dans lesquels il est autrement plus intéressant d'avouer que l'on ne sait pas plutôt que choisir une conclusion qui s'avérerait hasardeuse. Un des intérêts principaux du passage du graphe à la base de règles est justement de pouvoir traiter chaque règle séparément en s'assurant de leur pertinence. Dans [Zighed *et al.*, 1992], on décide qu'un règle est valable si elle est suffisamment précise (taux d'erreur en resubstitution minimum) et générale (effectif couvert par la règle supérieur à un seuil fixé par l'utilisateur).

Avec l'intensité d'implication, nous disposons d'un test statistique qui nous permet de valider un règle. Dès lors, il nous suffit de choisir la classe $y_{k'}$ qui maximise l'équation 6.4 et de conclure à la pertinence de la règle si elle vérifie la condition décrite dans l'équation 6.1.

Règle à conclusions "différent de ..."

Dans les problèmes à plus de deux classes, nous pouvons même affiner notre prise de décision en éditant des règles pour lesquelles nous concluons l'absence significative d'une classe ou d'un groupe de classes. En effet, le raisonnement étant basé sur la notion d'exemples et contre-exemples, en effectuant des regroupements judicieux, nous pourrons facilement les détecter.

Pour illustrer nos propos, nous allons prendre l'exemple de la base de données zoologiques. Le but de l'induction ici est de discriminer 7 classes d'animaux à partir de 15 attributs booléens et 2 numériques. Nous avons utilisé la méthode CART [Breiman *et al.*, 1984], l'arbre correspondant est représenté dans la figure 6.3.

Il apparaît que dans le sommet terminal le plus à gauche, il est difficile de trancher entre les classes F et G. D'un autre côté, on constate intuitivement l'absence significative des classes A, B, C, D et E.

Test d'équivalence distributionnelle Une manière simple de détecter de telles situations est d'utiliser un test du χ^2 d'équivalence distributionnelle. Ce test, dont l'hypothèse nulle consiste à vérifier si la distribution des classes est identique dans le sommet initial et le sommet où l'on extrait la règle (Table 6.3), permet ainsi de signaler les modifications de structures dans les différents sous-ensembles décrits par les prémisses. Un tel test est par exemple utilisé dans [Clark et Niblett, 1989] pour stopper la spécialisation des règles en induction. Son expression est assez simple

$$C = n \times n_{.l} \sum \frac{(\frac{n_k}{n} - \frac{n_{kl}}{n_{.l}})^2}{n_{k.} + n_{kl}}$$

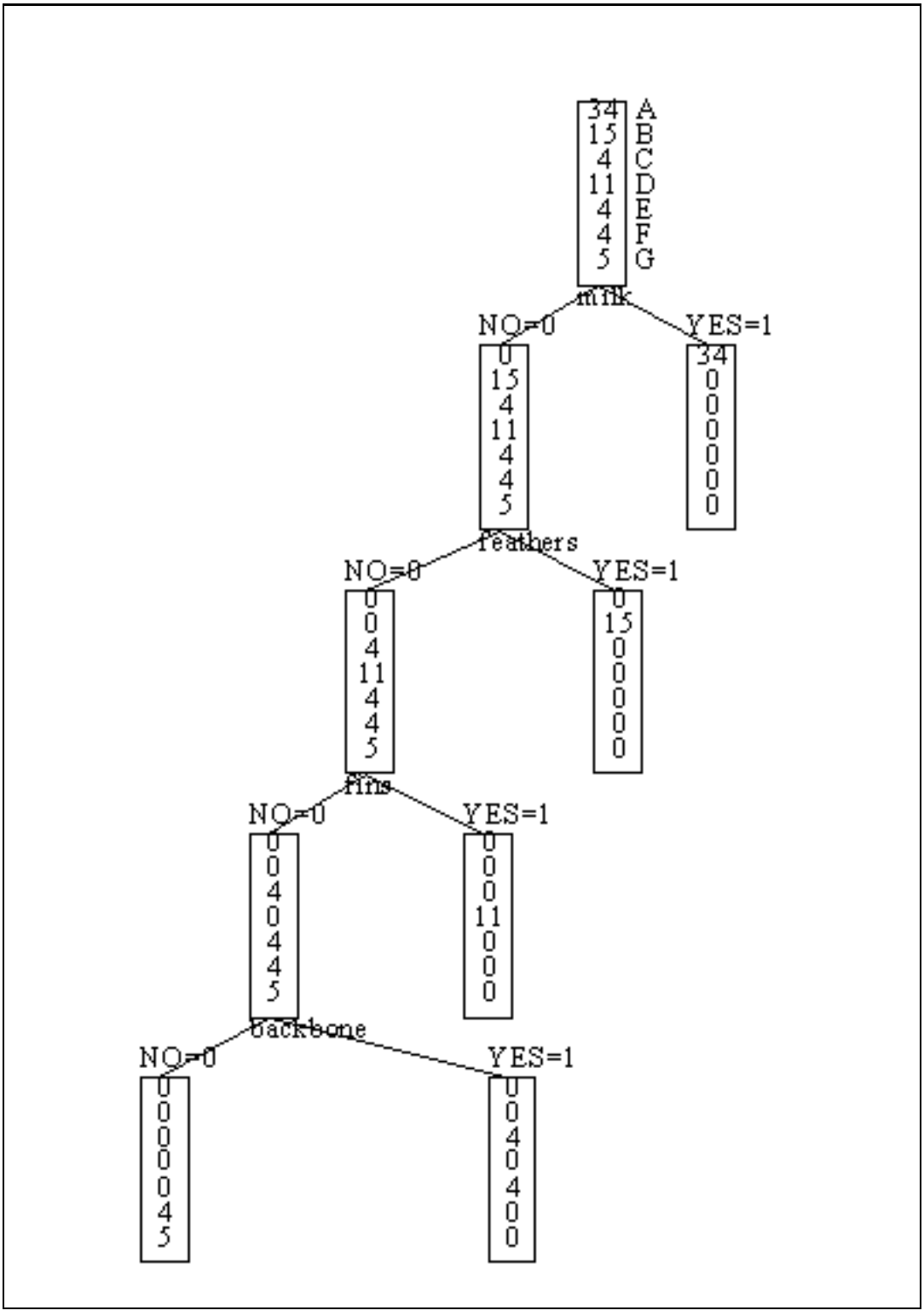

FIG. 6.3 – *Quelle est la conclusion la plus pertinente dans le sommet terminal le plus à gauche ?*

Classe	Initial	Terminal
A	34	0
B	15	0
C	4	0
D	11	0
E	4	0
F	4	4
G	5	5

TAB. 6.3 – *Distribution des classes dans la base Zoo (Sommet initial et terminal)*

Classe	Initial	Terminal
A&B&C&D&E	68	0
F&G	9	9

TAB. 6.4 – *Distribution des classes après regroupement dans la base Zoo*

Elle suit une loi de χ^2 à $(K-1)$ degrés de liberté. Pour un niveau de confiance α, nous décidons qu'il y a une différence significative de distributions des classes si

$$C \geq \chi^2_{1-\alpha}(K-1)$$

où $\chi^2_{1-\alpha}(K-1)$ est lue dans la table de distribution de la loi de χ^2 au point de pourcentage $1-\alpha$.

Dans notre exemple de la table 6.3, nous obtenons

$$C = 34.95$$

que l'on doit comparer avec $\chi^2_{0.95}(6) = 12.59$. Pour un niveau de confiance de 0.05, nous concluons qu'il y a bien une différence significative de distributions des classes entre le sommet initial et le sommet terminal. Le risque critique associé au test est ici de $P(\chi^2 \geq 34.95) = 1.13 \times 10^{-6}$.

Désignation des classes significativement absentes Cette première étape est certes très intéressante mais elle ne nous permet pas de détecter quelles sont les classes dont l'absence est significative. Pour ce faire, il est nécessaire d'effectuer les regroupements idoines pour travailler sur une représentation à deux classes. Le tableau initial de distribution des effectifs (table 6.3) devient (table 6.4).

Si nous ré-appliquons le test d'équivalence distributionnelle, le nombre de degrés de liberté est réduit à 1 alors que la valeur de C ne change pas. La valeur critique est $\chi^2_{0.95}(1) = 3.84$, et

le risque critique du test devient 7.16×10^{-10}. Nous constatons que le résultat est plus tranché que dans le cas à sept classes.

Ce passage à un problème à deux classes, avec un schéma "exemples" - "contre-exemples" nous suggère également l'utilisation de l'intensité d'implication pour établir l'absence significative d'un groupe de classes. L'application du test (cf. équation 6.1) ici nous permet bien de conclure à l'implication sur les classes F ou G ($3.5 \times 10^{-4} \leq 0.05$) : les classes A, B, C, D et E sont significativement absentes du sous-ensemble décrit par la prémisse.

Nous pouvons maintenant enrichir l'expression de la conclusion des règles en adoptant des négations de désignations de disjonctions de classes. Dans

$$\text{Si Pr}\textit{émisse Alors Conclusion}$$

Conclusion peut être de la forme

$$Classe \neq \{A, B, C, D, E\}$$

Construction des sous-groupes L'exemple ci-dessus présente l'avantage d'être très pédagogique, les classes absentes ont toutes un effectif nul. Comment faire dans la pratique pour construire ces sous-groupes?

La première idée qui vient à l'esprit est de tester toutes les combinaisons possibles, le nombre de cas possibles à évaluer pour une partition en c super-classes est donné par le nombre de Stirling du deuxième ordre

$$S(K, c) = \frac{1}{K!} \sum_{i=0}^{c} (-1)^{c-i} C_c^i \, i^K$$

Dans le cas particulier de $c = 2$ qui nous intéresse ici

$$S(K, 2) = 2^{K-1} - 1$$

Dans les problèmes réels, le nombre de classes est assez limité. Pour $K = 10$ par exemple, nous aurions 511 permutations à évaluer. Lorsque le nombre de classes augmente, cette stratégie est évidemment trop pénalisante en temps de calcul.

La deuxième méthode la plus simple consiste à adopter une démarche "hill-climbing". Nous commençons avec un groupe de classes à exclure vide. Nous excluons alors pas à pas les classes qui permettent de maximiser localement l'intensité d'implication, cela jusqu'à ce qu'il y ait $K-1$ classes dans le groupe des exclus. On choisit alors la partition qui aura maximisé notre mesure. On vérifie si elle est significative au regard du test d'absence des contre-exemples, on accepte la règle le cas échéant.

Bien entendu, ce type d'algorithme s'expose aux dangers inhérents à l'optimisation locale, on pourrait l'améliorer en utilisant des heuristiques telles que les algorithmes génétiques ou le recuit simulé.

Classe	*Etp.1*	*Etp.2*	*Etp.3*	*Etp.4*	*Etp.5*	*Etp.6*
A	**0.981**	-	-	-	-	-
B	0.833	**0.997**	-	-	-	-
C	0.373	0.988	0.9974	**0.9994**	-	-
D	0.724	0.995	**0.9991**	-	-	-
E	0.373	0.988	0.9974	**0.9994**	**0.9996**	-
F	0.008	0.457	0.7401	0.8665	0.8974	**0.9218**
G	0.000	0.307	0.6030	0.7691	0.8146	0.8528

TAB. 6.5 – *Evolution de l'intensité d'implication à mesure que l'on exclut une classe de la conclusion*

Nous recensons dans le tableau 6.5, les 6 étapes qui permettent d'aboutir à la partition en deux groupes du sommet terminal construit sur la base zoologique. Les chiffres indiquent l'intensité d'implication résultant de l'exclusion de la classe à l'étape i. Les classes effectivement exclues car maximisant l'intensité d'implication sont indiquées en gras.

Nous retrouvons bien en utilisant cette méthode la partition des classes en (A,B,C,D,E) et (F,G) puisque l'optimum est à l'étape 5 avec exclusion de la classe E.

6.4 Extraction de règles dans les graphes d'induction

Nous disposons maintenant des outils nécessaires pour assigner une conclusion à une règle et évaluer sa pertinence. Dans cette section, nous allons nous concentrer sur quelques méthodes d'extraction de règles à partir d'arbres.

6.4.1 L'extraction "classique" : parcours du graphe jusqu'à un sommet terminal

La méthode la plus simple [Quinlan, 1990] pour extraire les règles d'un graphe consiste à "suivre" les chemins menant aux sommets terminaux. Chaque proposition est construite à l'aide des tests sur les noeuds, et on affecte une classe à chaque feuille.

Sur l'exemple de la discrimination des Iris dû à [Fisher, 1936], nous pouvons voir la conversion d'un arbre de décision très simple 6.4 en trois règles :

$$Si\ (0 \leq PetalWidth \leq 0.8)\ Alors\ Iris = Iris\_Setosa$$
$$Si\ (PetalWidth \geq 0.8)\ et\ (0 \leq PetalWidth \leq 1.75)\ Alors\ Iris = Iris\_Versicolor$$
$$Si\ (PetalWidth \geq 0.8)\ et\ (PetalWidth \geq 1.75)\ Alors\ Iris = Iris\_Virginica$$

Généralement, les classifieurs d'obédience intelligence artificielle "forcent" la décision, même s'il est impossible de choisir entre deux classes sur un noeud terminal. [Cestnik *et al.*, 1987b] par

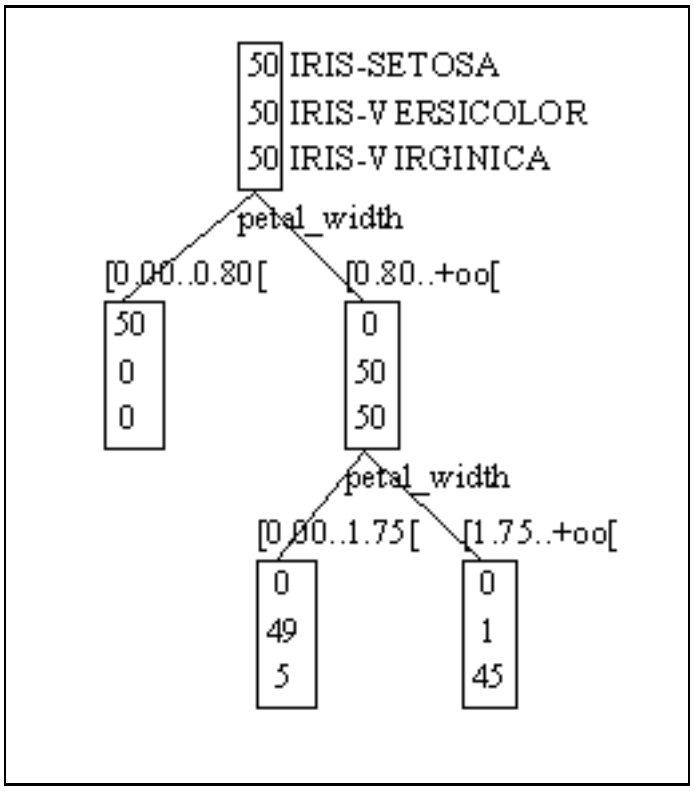

FIG. 6.4 – *Partition de la base des Iris en trois groupes*

exemple préconisent le choix aléatoire des classes lorsqu'il y a équidistribution.Nous avions discuté tantôt de l'opportunité de la prise de décision selon le domaine de l'étude que l'on réalise. Nous verrons dans la section suivante que l'on peut encore mieux exploiter l'information extraite par les graphes d'induction toujours en utilisant un procédé de validation des partitions représentées par les prémisses.

6.4.2 L'extraction par "validation" : la recherche des "formes" les plus pertinentes

Graphes et arbres d'induction définissent une série de catégorisations de la population [Rakotomalala et Chettouh, 1996]. Ces définitions ne se limitent pas aux noeuds terminaux, chaque noeud l représente un sous-ensemble d'individus décrits par la prémisse π_l. Nous voulons établir si la présence d'une classe y_k y est suffisamment forte pour que nous puissions extrapoler en définissant la règle $[R_t = (\pi_l, y_k)]$ dans la population totale. Dans la mesure où nous disposons d'un test qui nous permet d'évaluer la significativité des "formes" mises en évidence par la présence ou l'absence de telle ou telle classe, nous proposons de tirer les règles non plus à partir uniquement des sommets terminaux mais plutôt à partir de tout sommet non-initial du graphe. La validation nous permet ainsi de sélectionner les règles pertinentes. En modulant

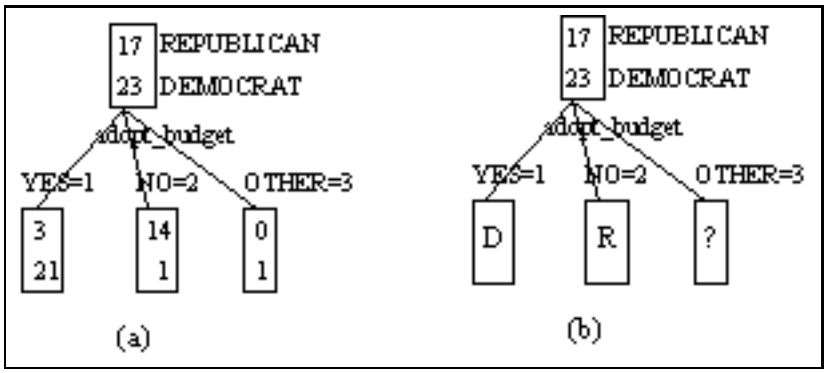

FIG. 6.5 – *En (a) figure le graphe à un niveau construit sur le fichier des Votes au Congrès. En (b) le graphe de décision après validation des sommets.*

le risque critique, nous pouvons définir plusieurs bases de règles homogènes du point de vue du test. Le graphe d'induction devient un graphe de décision où chaque conclusion est validée statistiquement.

Contrairement à la stratégie décrite précédemment (cf. 6.4.1), cette méthode permet un choix sélectif des sommets, les feuilles issues d'un même noeud peuvent être traitées différemment. Lorsqu'une partition engendre des sommets enfants de qualités décisionnelles très inégales, il serait dommage de nous priver de certaines règles sous prétexte que les noeuds frères sont de mauvaise qualité. C'est le raisonnement que peut adopter par exemple l'élagage lorsque le taux d'erreur moyen est mauvais par rapport au sommet père.

Avec ce système, deux sommets situés sur le même chemin peuvent engendrer deux règles concomitantes, menant à la même conclusion. Le choix de la règle à déclencher, et ceci reste applicable dans le cadre de la fusion de règles provenant de plusieurs bases de données similaires, pourrait alors se faire sur la base de l'intensité d'implication : on activera en classement celle qui présentera la causalité la plus forte au sens du test.

De plus, il est alors tout à fait possible, maintenant que la base est homogène au sens du test que nous avons défini, d'utiliser des méthodes de simplification symboliques pour éliminer les éventuelles redondances [Rabaseda *et al.*, 1996a]. Rappelons que dans de telles méthodes, on considère que les règles ont un poids identique et sont traitées sur un même pied d'égalité.

Nous avions utilisé un exemple simple à partir de la base des Votes au congrès dans [Rakotomalala et Zighed, 1996]. Nous représentons dans [Figure 6.5(a)], l'arbre construit sur 40 individus (la base comporte 435 observations).

Nous pouvons en extraire trois règles, en adoptant l'assignation à la classe majoritaire :

R_1 : Si (AdoptBudget=Yes) Alors Democrate
R_2 : Si (AdoptBudget=No) Alors Republicain
R_3 : Si (AdoptBudget=Other) Alors Democrate

Si les règles R_1 et R_2 semblent justifiée, R_3 est franchement suspecte. Afin de nous en assurer, nous consignons dans le tableau 6.6 : le taux d'erreur en resubstitution de la règle, l'intensité d'implication associée et le taux d'erreur en validation sur les 395 individus restants.

Règles	Tx*Resubs.*	*Int.Impl.*	*TxValid*
R_1	0.125	0.991^{**}	0.08
R_2	0.067	0.998^{**}	0.18
R_3	0.0	0.346	0.40

TAB. 6.6 – *Taux d'erreur en resubstitution, intensité d'implication (** règle valide à 0.01) et taux d'erreur en validation*

Les résultats montrent que la règle R_3 est effectivement fragile en prédiction, le comportement le plus sage serait de s'abstenir de prédire dans ce type de situation.

6.5 Constitution d'une base de connaissances : Traitements associés

Une des premières raisons du passage du graphe à une base de règles est peut être une meilleure lisibilité de la connaissance produite. Lorsque le graphe est complexe et enchevêtré, la construction des règles permet de le mettre à plat en décomposant au mieux la description des différentes partitions [Corlett, 1983]. La deuxième raison, que nous avions déjà invoquée en introduction, était l'utilisation du classifieur dans les systèmes experts.

La troisième raison qui nous intéresse ici est la possibilité d'appliquer des opérations sur la base de règles issue des graphes afin d'améliorer ses performances et/ou diminuer sa complexité. Ce nouveau système de représentation permet des manipulations interdites avec la structure des graphes. On peut ainsi manipuler des règles les unes indépendamment des autres alors que cela n'était pas possible notamment dans les arbres où les sommets enfants étaient jugés en groupe face à leur sommet père lors de l'élagage.

Afin de remédier par exemple à l'incapacité des arbres à saisir les concepts en forme normale disjonctive, certains auteurs [Dietterich, 1990], plutôt que de passer aux graphes, plus riches, proposent l'utilisation de la simplification dans une deuxième phase de traitement de la base de règles afin d'éliminer les propositions redondantes. [Quinlan, 1993a], avec C4.5 Rules, pense même que l'on peut améliorer dans le même temps la compacité et les performances du classifieur en éliminant les propositions qui sont induites par le bruit de l'échantillon d'apprentissage. Il utilise alors un algorithme numérique qui modifie les propriétés du classifieur sur la base d'apprentissage. [Rabaseda *et al.*, 1996a] proposent en revanche une simplification symbolique dans laquelle la base de règles garde exactement les même performances sur le fichier d'apprentissage.

Nous présentons dans leurs grandes lignes ces différents algorithmes de simplification. Mais auparavant, nous allons essayer de trouver un "bon" indicateur de complexité de la base de règles afin de juger de leurs performances.

6.5.1 Complexité d'une base de règle

La manière la plus simple de quantifier la complexité d'un base de règles est de compter le nombre de règles qui la composent [Rakotomalala *et al.*, 1997a]. Cette méthode est néanmoins très critiquable car elle ne tient pas compte du fait que certaines règles sont plus complexes que d'autres.

Dans [Clark et Boswell, 1991], les auteurs proposent de faire la somme totale du nombre de propositions rattachées à chaque règle. Ainsi, on tient compte à la fois du nombre de règles et de leurs longueurs respectives.

Notre sentiment est que ces deux valeurs doivent être assez liées, au moins en ce qui concerne les graphes d'induction. En effet, lorsque les règles contiennent beaucoup de propositions, elles sont très spécialisées et ne couvrent qu'une très petite fraction de l'échantillon d'apprentissage. De fait, il faut un très grand nombre de règles pour couvrir l'ensemble de la base à étudier.

Afin de vérifier cette assertion, nous avons effectué des tests sur nos 15 bases exemples. Le protocole utilisé a été le suivant : pour chaque base nous avons effectué 20 tirages bootstrap sur lesquels nous avons construits des graphes à l'aide de la méthode SIPINA. Nous avons ensuite calculé un coefficient de corrélation de rangs de Spearman entre le nombre de règles générées par la méthode et le nombre de propositions correspondantes. Les résultats sont relevés dans le tableau 6.7.

Ces résultats sont édifiants. Au moins en ce qui concerne les graphes d'induction, le nombre de règles extraites est extrêmement corrélé avec la somme totale de propositions contenues dans la base de connaissances. On aura remarqué cependant lors de nos expérimentations que la variabilité du nombre de règles était très faible face à celle du nombre de propositions.

Peut-on pour autant les interchanger dans l'analyse de la réduction de la complexité des bases lorsque l'on leur appliquera un algorithme de simplification? Nous sommes moins catégoriques ici. En effet, tout dépend de la nature de l'algorithme utilisé. S'il s'agit d'un simple système d'élagage qui se contente d'analyser les règles unes à unes et d'exclure les propositions peu pertinentes, il est clair que le nombre de règles ne changera pas alors que le nombre de propositions diminuera certainement. En revanche, s'il s'agit d'une méthode visant à simplifier la base dans sa globalité, nul doute qu'elle sera en mesure d'exclure également les règles redondantes après leur traitement. On peut penser qu'ainsi la relation entre ces deux indicateurs restera assez stable.

Néanmoins, afin d'éviter toute polémique, nous avons décidé d'utiliser la somme du nombre de propositions par règles pour appréhender la réduction de la complexité d'une base dans notre

Base	**Coef. Spearman**
automobile data	0.9426
breast-cancer wisconsin	0.9482
car evaluation	1
computer hardware	1
credit scoring	0.9749
flags	0.9523
hepatitis	0.9639
ionosphere	0.8152
iris	1
lung cancer	0.9578
pima diabetes	0.9578
vote congress	1
wave	0.8991
wine	0.9193
zoo	1

TAB. 6.7 – *Coefficient de corrélation entre le nombre de règles et le nombre de propositions dans la base de connaissances extraite des graphes*

analyse. De plus, sa plus grande variance montre qu'elle est plus sensible aux changements.

6.5.2 Simplification d'une base de règles à l'aide d'un algorithme symbolique

Dans [Rabaseda, 1996] sont recensées quelques méthodes symboliques de réduction de règles. Certaines sont fondées sur des algorithmes génétiques [Grange *et al.*, 1995], d'autres sont inspirées de la minimisation des portes logiques dans les circuits éléctroniques [Brayton *et al.*, 1992].

Nous avions énormément travaillé sur ce dernier type d'algorithme. Sa rapidité et sa simplicité nous avaient séduits, nous pensions que l'on pouvait l'adapter assez facilement à la simplification de prémisses en logique propositionnelle d'ordre O^+ (attribut-valeur). Hélas, les limitations inhérentes à cette méthode se sont révélées rédhibitoires, le codage disjonctif complet des variables entraînant une perte d'information telle que l'on obtenait par la suite des solutions contenant des propositions contradictoires.

Toutefois, nous nous sommes beaucoup inspirés de ces résultats pour élaborer un algorithme de force brute [Rakotomalala, 1995a] qui permet de trouver l'expression optimale d'une base de règles. Il comporte deux phases distinctes :

- réduction de la redondance dans chaque règle;
- génération de toutes les règles possibles, celles qui s'avèrent être les plus générales (contenant les prémisses les plus courtes) et recouvrant une ou plusieurs règles de la base de connaissances sont introduites et éjectent les précédentes. L'exploration de l'espace des solutions a été considérablement réduit grâce à quelques "astuces", notamment en utilisant le fait qu'il ne peut y avoir de règles portant deux propositions sur la même variable.

Afin d'illustrer son fonctionnement, nous donnons un exemple simple construit sur le fichier des cancers du sein (breast cancer - wisconsin). Le graphe résultant est représenté dans la figure 6.6 . Les règles produites en adoptant une assignation à la majorité simple et une extraction aux sommets terminaux sont :

R_1 : Si $(0 \leq ucellsize < 2.5)$ *Alors Benin* $\{12, 417\}$
R_2 : Si $(ucellsize \geq 2.5)$ *et* $(0 \leq ucellshape < 2.5)$ *Alors Benin* $\{5, 18\}$
R_3 : Si $(ucellsize \geq 2.5)$ *et* $(ucellshape \geq 2.5)$ *et* $(0 \leq ucellsize < 4.5)$ *Alors Malin* $\{52, 18\}$
R_4 : Si $(ucellsize \geq 2.5)$ *et* $(ucellshape \geq 2.5)$ *et* $(ucellsize > 4.5)$ *Alors Malin* $\{172, 5\}$

Les chiffres entre accolades indiquent le nombre d'individus couverts par la règle dans chaque classe.

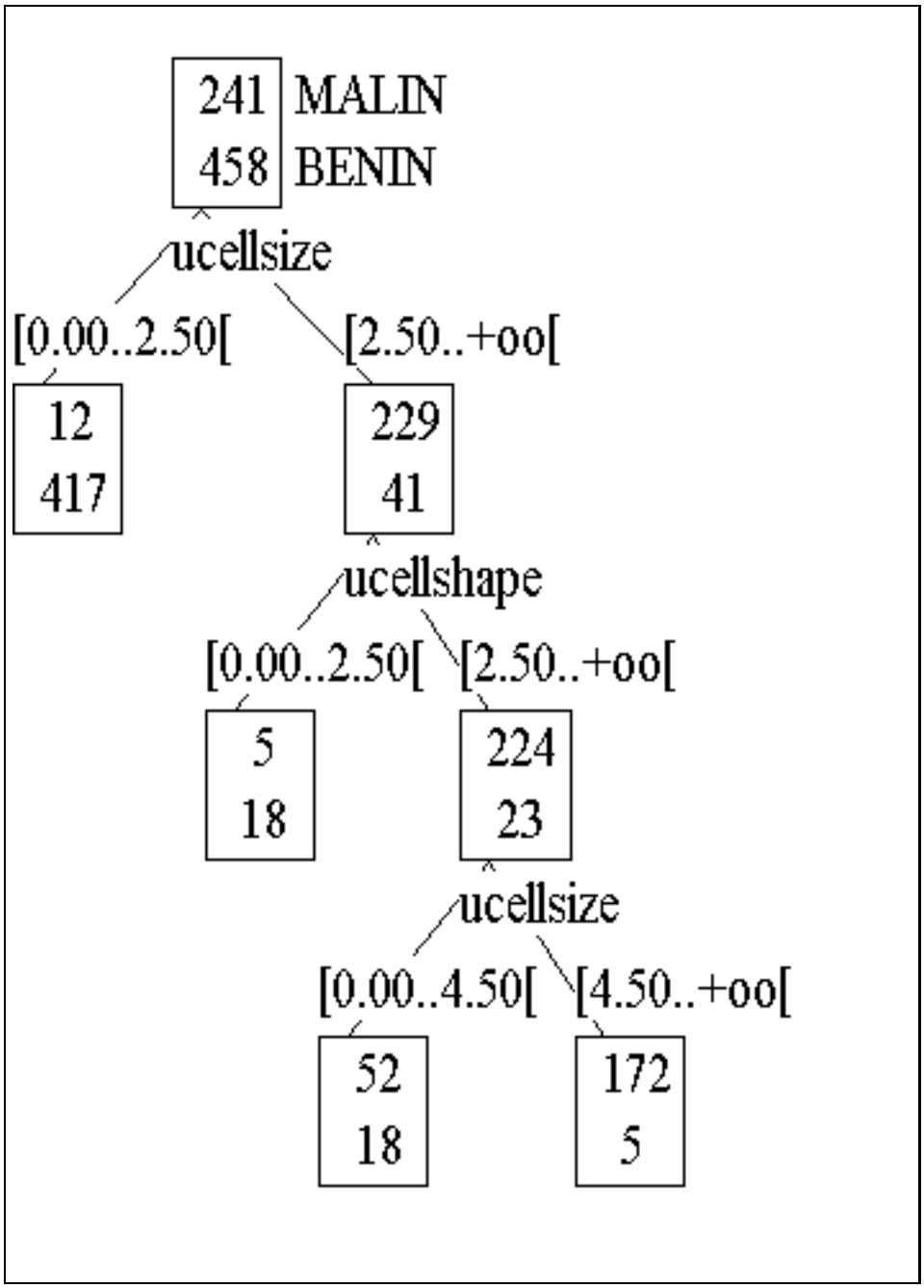

FIG. 6.6 – *Graphe construit sur le fichier des Cancers du sein*

La première phase consiste d'abord à enlever les redondances dans chaque règle. On remarque ici que l'attribut 'ucellsize' revient deux fois dans les règles R_3 et R_4. On procède à une réécriture plus compacte de chaque règle dont la couverture n'est pas modifiée.

$$\begin{array}{|l|}\hline R_3 : \text{Si } (2.5 \leq ucellsize < 4.5) \; et \; (ucellshape \geq 2.5) \; Alors \; Malin \; \{52, 18\} \\ R_4 : \text{Si } (ucellsize \geq 4.5) \; et \; (ucellshape \geq 2.5) \; Alors \; Malin \; \{172, 5\} \\ \hline \end{array}$$

La deuxième phase, plus puissante, consiste alors à générer toutes les règles possibles à partir des attributs présents dans la base de connaissances, puis ne garder que l'ensemble de règles qui possède exactement la même couverture mais avec une spécification plus concise. Après traitement, nous obtenons trois règles :

$$\begin{array}{|l|}\hline R_a : \text{Si } (0 \leq ucellsize < 2.5) \; Alors \; Benin \; \{12, 417\} \\ R_b : \text{Si } (0 \leq ucellshape < 2.5) \; Alors \; Benin \; \{9, 403\} \\ R_c : \text{Si } (ucellsize \geq 2.5) \; et \; (ucellshape \geq 2.5) \; Alors \; Malin \; \{224, 23\} \\ \hline \end{array}$$

Ces résultats amènent quelques remarques que nous formulons ci-après :

- R_a est identique à R_1, il n'y a pas dans l'espace de description une spécification plus compacte de cette règle;
- R_b est une réexpression de R_2 tenant compte de l'information apportée par R_1;
- R_c est un condensé de l'union des individus couverts par R_3 et R_4;
- on remarque que les règles sont plus concises et couvrent en moyenne une fraction plus élevée de l'échantillon d'apprentissage, augmentant ainsi leur fiabilité;
- on notera également que cette simplification n'a de sens que si les règles sont ordonnées, regroupant au moins celles menant à des conclusions identiques.

Si cette méthode a donné d'excellents résultats sur de petits fichiers, avec peu d'attributs, elle s'est révélée impraticable dès que le nombre de variables et les modalités correspondantes ont augmenté. Nous l'avions quand même gardée, car testant toutes les occurrences possibles, nous sommes sûr qu'elle est optimale.

Toutes les méthodes de cette sous-section ont pour point commun de donner la même importance à toutes les règles. En ce sens, il est alors très important d'avoir une base homogène au sens d'un critère que l'on se fixera. L'utilisation d'un test de validité comme barrière à l'entrée permet d'exclure les règles de qualité douteuse pouvant altérer les performances de l'ensemble.

6.5.3 Simplification d'une base de règles à l'aide d'un algorithme numérique

Il nous faut garder à l'esprit que la base de connaissances est issue d'un processus d'apprentissage. On peut se demander si en re-traitant les règles extraites des graphes à l'aide d'un algorithme numérique, nous ne pourrions pas trouver des expressions intrinsèquement meilleures au sens de la précision et de la généralité.

La méthode la plus connue est certainement C4.5 rules [Quinlan, 1993a] qui est très populaire au sein de la communauté des chercheurs travaillant sur les arbres de décision. Elle comporte deux étapes distinctes que nous explicitons dans les deux sous-sections suivantes.

Simplification par élagage

La première étape consiste à tronquer les règles en enlevant les propositions les moins intéressantes. L'auteur essaie d'optimiser, en utilisant un algorithme glouton, la borne basse du taux de succès (ou ce qui revient au même, la borne haute du taux d'erreur). Cette démarche qui s'apparente à du post-élagage [Michalski *et al.*, 1986] est exactement l'inverse des techniques de spécialisation [Clark et Boswell, 1991] dans lesquelles on essaie de construire les règles par adjonction de propositions.

Cette phase peut être améliorée de trois manières différentes :

- utilisation de mesures plus performantes du point de vue de la recherche des expressions les plus générales pour des taux d'erreur apparents identiques [Rakotomalala *et al.*, 1997a];
- utilisation d'un échantillon test sur lequel nous minimisons le taux d'erreur, ce qui s'apparente à la méthode CART dans la construction des arbres de décision [Breiman *et al.*, 1984];
- utilisation de méthodes exploratoires plus puissantes dans la généralisation des règles, l'algorithme glouton pêchant parfois devant des optima locaux [Venturini, 1994].

Minimisation du coût de description des données

Cette première étape permet d'écourter les règles, il arrive alors que certaines soient redondantes. La deuxième phase vise à exprimer la base de règles de la manière la plus concise possible.

[Quinlan, 1993a] propose une approche fondée sur la description minimale des données que nous avions déjà abordé plus haut (cf. 6.2.2.0). Connaissant le coût de description de chaque règle, le coût de description d'un ensemble de règles est tout simplement la somme du coût de description de chacune de ses composantes additionnée au coût de description des faux positifs et des faux négatifs, auquel on soustrait $\log(nombre\_de\_règles\,!)$ puisqu'elles peuvent être envoyées dans n'importe quel ordre.

Pour un ensemble de règles S_k, de cardinal s_k, menant à la conclusion y_k, le coût de description s'écrit

$$C(S_k) = \sum_{i=1}^{s_k} C_{Théorie}(R_i) + C_{Exceptions}(S_k)$$

Les "faux positifs" ici représentent l'ensemble des individus couverts par S_k ne portant pas l'étiquette y_k, et les "faux négatifs", les individus portant l'étiquette y_k non-couverts par S_k.

L'algorithme consiste tout simplement à détecter pour chaque classe conclusion y_k le groupe de règles élaguées S_k^* qui minimise le coût de description global $C(S_k)$. On peut encore une fois utiliser une heuristique gloutonne, mais l'auteur penche pour un algorithme de recuit simulé inspiré des formulations de [Press *et al.*, 1988] dans les fameuses séries de "Numerical Recipes". Dans nos tests, nous y avons substitué un simple algorithme de percolation qui enlève itérativement les règles de S_k.

On reproche souvent à C4.5 rules de ne pas offrir une couverture identique à l'ensemble de départ [Rabaseda, 1996]. Il arrive que certains individus ne soient couverts par aucune règle. Dans ce genre de situation, l'auteur définit une conclusion par défaut qui est tout simplement l'affectation à la classe la plus représentées dans l'ensemble des individus non-couverts en apprentissage. Nous verrons dans ce qui suit que cette solution qui ressemble plus à une béquille qu'autre chose se révèle extrêment performante dans la pratique.

6.5.4 Comparaison sur quelques fichiers exemples

L'objectif n'est pas tant de comparer ces deux familles d'algorithmes pour déterminer quel est le meilleur, mais plutôt d'essayer de qualifier leur comportement sur des données réelles. D'une part, nous avons un algorithme symbolique qui préserve les qualités de prédiction du classifieur sur l'échantillon d'apprentissage en adoptant une expression plus concise; d'autre part, nous avons un algorithme numérique qui cherche également une expression plus précise, mais en modifiant les qualités prédictives du classifieur en apprentissage. Un des principaux enjeux est d'observer l'effet de la simplification sur un échantillon test.

Dans les tableaux (6.8, 6.9 et 6.10), nous avons relevé avant et après simplification : le nombre de propositions (*Prop...*), le taux de succès en resubstitution (*App...*), et le taux de succès en validation (*Val...*). Les données en parenthèses indiquent l'effectif de l'échantillon utilisé. Devant la lenteur récurrente de l'exploration exhaustive des solutions (cf. 6.5.2), nous n'avons pu tester que quatre bases de données qui présentaient l'avantage de contenir peu d'attributs prédictifs, nous n'effectuerons donc pas d'étude statistique sur les performances comparées des différentes stratégies.

Le protocole utilisé dans cette expérimentation a été de construire un arbre de décision à l'aide de C4.5 [Quinlan, 1993a]. De la base de règles extraite, nous en avions généré deux à l'aide

Fichier	*Prop*	*Prop.AS*	*Prop.C4.5*
Breast (150)	21	17	8
Pima (150)	28	28	14
Wine (90)	53	20	18
Hepatitis (50)	13	20	5

TAB. 6.8 – *Nombre de propositions dans la base de règles*

Fichier	*App.*	*App.AS*	*App.C4.5*
Breast (150)	98	98	97
Pima (150)	84	84	83
Wine (90)	94	94	91
Hepatitis (50)	88	88	87

TAB. 6.9 – *Taux de succès sur le fichier d'apprentissage*

de la simplification symbolique et de C4.5 rules. Nous avons par la suite mesuré leur complexité et leurs performances en classement à la fois sur le fichier d'apprentissage et de validation.

Les résultats nous ont laissé perplexes et amènent plusieurs remarques :

- notre algorithme symbolique peut parfois engendrer des bases de règles encore plus complexes (complexité quantifiée en terme de nombre de propositions) que la base originelle (Table 6.8, fichier Hepatitis). Ce résultat contradictoire se comprend aisément. En effet, parmi les règles générées, il en existe qui couvrent partiellement plusieurs autres, de fait on ne peut pas exclure ces dernières car il y aurait des individus non-couverts. De fait malgré que l'on explore dans sa totalité l'espace des solutions, l'ordre avec lequel nous les étudions influe sur la qualité du résultat. Cette propriété et la lenteur rédhibitoire de cette stratégie nous ont d'ailleurs poussé à l'abandonner, nous nous en servons uniquement maintenant comme méthode témoin.

- C4.5 rules réduit de manière significative la complexité des bases de règles (Table 6.8). Les études que nous avons menées sur les autres fichiers exemples montrent de manière signi-

Fichier	*Val.*	*Val.AS*	*Val.C4.5*
Breast (549)	92	92	93
Pima (618)	73	73	74
Wine (88)	86	86	93
Hepatitis (62)	79	87	87

TAB. 6.10 – *Taux de succès sur le fichier de validation*

ficative cette diminution. Par exemple sur les fichiers des Ondes de Breiman, le classifieur est passé de 365 à 33 propositions. La règle par défaut, la classe la plus fréquente parmi les individus non-couverts, joue sans conteste un rôle très important. A la limite, nous pourrions exclure complètement l'ensemble de règles désignant y_k, réduire les sous-ensembles de règles menant aux autres classes, puis assigner à la règle par défaut la conclusion y_k.

- en resubstitution (Table 6.9), notre base simplifiée par la méthode symbolique classe d'une manière identique tous les individus du fichier d'apprentissage. C4.5 rules voit son taux de succès en resubstitution diminuer. Ce qui était prévisible, le bruit évacué laisse croire des performances dégradées sur le fichier d'apprentissage.

- en validation (Table 6.10), C4.5 rules, malgré une réduction drastique de la complexité du classifieur, garde un niveau de performances comparable aux autres. La simplification symbolique procède de manière identique que le classifieur originel, sauf dans le cas où il a généré un plus grand nombre de règles.

Ces résultats semblent militer en faveur de la simplification fondée sur un algorithme d'optimisation numérique. Ils montrent surtout que le changement de représentation pour retraiter le classifieur amène de gros avantages : réduction de complexité, et éventuellement, meilleure résistance au bruit.

6.6 Conclusion et perspectives

La possibilité du passage aux règles est un des meilleurs atouts des graphes, ce qui le rend dans les faits très pratique pour constituer par exemple les bases de connaissances de systèmes experts. Notons cependant que ce n'est pas un processus trivial. Différentes options peuvent être mises en oeuvre pour s'assurer de la qualité des règles. Mieux encore, on peut y apporter des amélioration en appliquant un algorithme numérique qui profite du changement de système de représentation. Enfin, l'adoption d'un système de représentation universel permet de fusionner des connaissances d'origines différentes : d'autres algorithmes d'induction, des règles d'expert...

La démarche inverse existe, [Michalski et Imam, 1994] ont travaillé sur le passage d'une base de règles à un arbre de décision, paradoxalement c'est la lisibilité du processus de décision qu'ils invoquent comme le principal argument de cette transformation.

Troisième partie

Innovations dans les graphes d'induction

Chapitre 7

Graphes d'induction

7.1 Introduction

Jusqu'ici nous utilisions indifféremment les termes "graphes" et "arbres" d'induction pour désigner le modèle d'apprentissage, sans vraiment nous poser la question de savoir s'il y avait une distinction conceptuelle entre ces deux terminologies. Du point de vue de la théorie des graphes, il est évident qu'un arbre est un cas particulier d'un graphe. En effet, un arbre est un graphe orienté sans cycle avec une racine où à chaque noeud non-terminal est assignée une variable $X(.)$ qui y induit une segmentation, avec des arcs sur lesquels sont portés les modalités x_i de $X(.)$. Le graphe inclut toutes ces propriétés, il y ajoute la possibilité pour un noeud d'avoir deux pères, ou ce qui revient au même, pour deux noeuds de posséder le même noeud enfant [27]. Enfin, classiquement, à chaque feuille du graphe (ou de l'arbre) est assignée la classe qui correspond à la décision lorsque l'on parcourt le graphe du sommet initial (la racine) à la feuille.

L'introduction de l'opération de fusion étend considérablement le pouvoir de représentation du modèle. C'est la principale motivation de son introduction en apprentissage ces dernières années [Oliveira, 1994]. Des concepts qui jusque là nécessitaient des arbres de grande taille pour en couvrir la définition, sont maintenant traduits à travers des graphes plus concis, solutionnant ainsi élégamment deux problèmes principaux qui ont monopolisé les chercheurs : la réplication des sous-arbres et la fragmentation des données.

Historiquement en France, les travaux relativement anciens sur les processus d'interrogation latticiels [Terrenoire, 1970] [Tounissoux, 1980] constituent les prémices de l'induction par graphes qui connut son apogée avec la méthode SIPINA (Système Interactif pour les Processus

27. cette avancée peut être vue comme une limitation, en effet pourquoi fixe-t-on à deux le nombre de pères que peut posséder un noeud? Ne peut-on pas envisager la fusion de trois noeuds ou plus? A notre connaissance, il n'existe pas de travaux autour de cette extension, le choix du nombre de feuilles à fusionner et une méthode d'exploration rapide semblent être les principaux enjeux dans ce cadre.

d'Interrogation Non Arborescent) [Zighed, 1985]. La construction de ce type de représentation était rendue possible par l'adoption d'une mesure de qualité sur la partition globale permettant la comparaison entre une opération de segmentation et une opération de fusion [Oliver, 1993].

Dans le monde anglo-saxon, l'émergence des graphes a suivi un itinéraire plutôt torturé. Le principal souci des auteurs a d'abord été d'essayer de trouver des solutions aux problèmes sus-cités. Les premiers travaux recensés sont les oeuvres de [Rivest, 1987] avec les "decision list" et de [Chou, 1988][28] avec les "decision trellis", qui constituent une généralisation des "decision pylon" de [Bahl *et al.*, 1989]. Tous ces systèmes de représentations sont des cas particuliers des graphes d'induction dont l'élaboration, en totale ignorance des travaux français, fût l'oeuvre de [Oliver et Wallace, 1991]. Ce dernier propose un algorithme en tout point semblable à celui de [Zighed, 1985], seule la mesure à optimiser diffère : il essaie de minimiser la longueur du message pour décrire le graphe. Dans un autre registre, d'autres chercheurs, toujours séduits par la puissance des graphes, pensent que l'on peut transformer un arbre de décision classique en graphe en usant d'opérations de simplification et de fusion entre séquences d'arbres analogues [Oliveira, 1994][Kohavi, 1994].

L'organisation de ce chapitre sera la suivante : dans un premier temps, nous décrirons les problèmes de la réplication et de la fragmentation avec les solutions afférentes en terme de graphes; dans un second temps, nous détaillerons successivement les algorithmes de dérivation des graphes à partir des arbres de décision et les algorithmes gloutons de constructions de graphes. Enfin, nous mènerons des comparaisons empiriques entre un graphe d'induction et un arbre de décision, totalement identiques, mis à part la possibilité d'effectuer des fusions. Nous conclurons alors.

7.2 Motivations du passage aux graphes

Chaque sommet du graphe d'induction décrit un sous-ensemble de la population. En particulier, les feuilles en induisent une partition disjointe. Pendant longtemps, la construction des arbres de décision a souffert du sur-apprentissage qui se traduisait par des sommets terminaux de taille exagérément faibles [Zighed *et al.*, 1992]. Avec CART et le développement des méthodes de recherche de la taille optimale [Breiman *et al.*, 1984][Wehenkel, 1993], on a trouvé une manière d'éviter cette sur-spécialisation sans véritablement se pencher sur son origine. Il était communément admis que le bruit en était le principal responsable.

Pourtant, même en l'absence de bruit dans les données, les arbres souffrent de problèmes qui hypothèquent véritablement leur efficacité, au moins en terme de concision.

28. auquel on attribue d'ailleurs la terminologie "decision graph" [Oliver *et al.*, 1992].

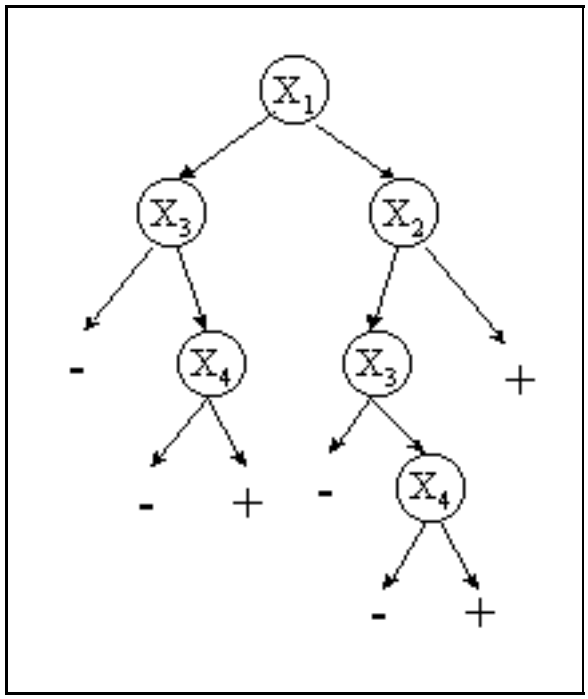

FIG. 7.1 – *L'arbre de plus petite taille pour traduire le concept* $f = x_1.x_2 + x_3.x_4$

7.2.1 La réplication des sous-arbres

Sans restreindre la portée de notre discours, nous allons nous placer dans le cas particulier où tous les attributs, y compris la variable à prédire, sont booléens. Par convention, nous décidons que les arcs positifs sont situés à droite du noeud.

Dans la figure 7.1, nous représentons le plus petit arbre de décision pour décrire la fonction booléenne

$$f = x_1.x_2 + x_3.x_4$$

Nous constatons que le deuxième concept $(x_3.x_4)$ nécessite deux séquences de sous-arbres identiques pour que la fonction soit complètement couverte. La raison de cette duplication est dans l'apparition de formes disjonctives dans le concept à décrire, elle a été identifiée par [Matheus et Rendell, 1989] et [Pagallo et Haussler, 1990].

Il y a principalement deux conséquences malheureuses à cette inefficience :

– la première est la taille gigantesque des arbres pour décrire des concepts finalement assez simples. Par exemple, le plus petit arbre pour décrire l'expression logique

$$g = x_1.x_2.x_3 + x_4.x_5.x_6 + x_7.x_8.x_9 \tag{7.1}$$

contient 39 noeuds et 40 feuilles [Quinlan, 1987c]. Ce qui pose d'énormes problèmes de stockage et de compréhensibilité lors de leur conversion en base de règles pour l'insertion

dans un système expert [Michie, 1987] : pour cet exemple, au lieu d'avoir simplement 3 règles, nous sommes obligés d'en gérer 40.

– cela veut dire également une partition exagérée de l'ensemble d'apprentissage. Toujours pour notre exemple ci-dessus, l'échantillon de départ devra être divisé en 40 sous-ensembles pour décrire chaque règle, alors que théoriquement trois sous-ensembles sont nécessaires. La construction d'un arbre exige donc un grand nombre d'observations.

Face à de tels problèmes, il y a plusieurs solutions qui ne sortent pas du cadre des arbres de décision. La première est l'oeuvre de [Pagallo et Haussler, 1990] qui proposent de construire des variables synthétiques booléennes à partir des attributs de départ [29]. En adoptant un processus itératif qui élabore incrémentalement des concepts, il est possible de capturer chaque conjonction de propositions. L'arbre final se présentera alors sous la forme de disjonctions de ces prémisses. La seconde est l'oeuvre de [Quinlan, 1993a], il se propose de construire normalement l'arbre puis de le convertir en base de règles. Ce changement de système de représentation autorise l'application d'autres opérateurs, notamment la simplification, qui permet de trouver l'expression optimale en terme de concision.

Certes, ces algorithmes ont fait leurs preuves, et permettent effectivement de réduire considérablement les bases de règles issues de l'apprentissage sur des exemples fabriqués et réels. Mais il reste qu'elles nécessitent un effectif élevé pour reconstruire complètement le concept avant de le simplifier. Pour l'exemple de l'équation 7.1, avec un niveau de bruit de 10%, [Wallace et Patrick, 1993] ont montré qu'il fallait un échantillon de 3000 individus pour en restituer la teneur. Lorsque le coût de l'acquisition des données est élevé, il est évident que ce type de solution est impraticable.

La solution que nous préconisons dans ce chapitre est l'élaboration des graphes d'induction. Connus sous le terme de "diagrammes booléen de décision" en logique [Oliveira et Sangiovanni-Vincentelli, 1995], ils couvrent en utilisant des expressions plus concises toutes les expressions booléennes que peuvent traduire les arbres [Oliver et Wallace, 1991]. De fait, il s'agit véritablement ici d'une généralisation qui répond de manière plus élégante au problème de la réplication..

Si nous reprenons l'arbre de la figure 7.1, nous représentons dans la figure 7.2 le graphe où l'on fusionne les séquences d'arbres répliquées. De fait, nous obtenons un modèle plus simple, 3 noeuds intermédiaires et 2 feuilles contre 5 noeuds et 7 feuilles, qui se traduit également par un effectif nécessaire plus faible pour apprendre le concept. Dans l'exemple de la fonction g (équation 7.1), 400 individus sont suffisants pour construire le graphe, là où 3000 étaient nécessaires pour créer l'arbre correspondant [Oliver, 1993].

En conclusion, les graphes, par rapport aux arbres classiques, proposent des expressions plus concises et peuvent couvrir des concepts relativement complexes avec des échantillons de petite

29. Nous étudierons plus en détail ce champ d'étude que constitue la construction de descripteurs (feature construction)

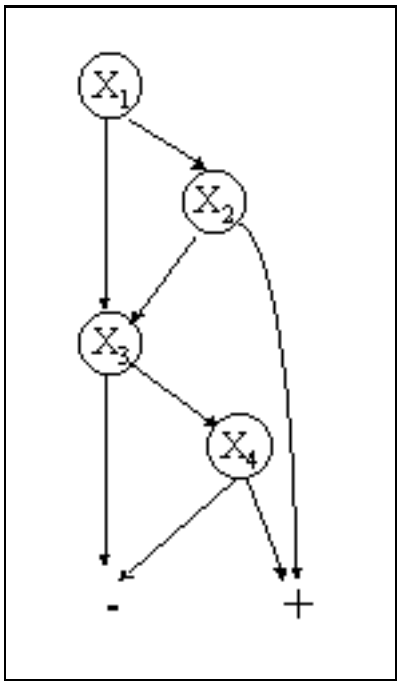

FIG. 7.2 – *Le graphe de plus petite taille pour traduire le concept* $f = x_1.x_2 + x_3.x_4$

taille.

7.2.2 La fragmentation des données

Cette faculté d'utiliser efficacement les échantillons est mise à contribution dans le deuxième sérieux problème qui hypothèque les arbres de décision : la fragmentation des données.

Lorsque les attributs prédictifs contiennent un grand nombre de modalités, l'éclatement sur un noeud occasionne un rapide fractionnement des données dans les sommets enfants. Cette subdivision répétée sur les sommets successifs d'un chemin contribue alors à produire des arbres très instables, avec des feuilles contenant des sous-échantillons de taille très réduite, compromettant la fiabilité du classifieur.

La solution la plus évidente a été la binarisation des attributs [Cheng *et al.*, 1988] [Cestnik *et al.*, 1987a] : les modalités de la variable prédictive sont regroupées en deux super-modalités qui s'accordent le mieux avec le pronostic de la variable à prédire. S'il y a des solutions optimales lorsque la variable à prédire ne prend que deux classes [Breiman *et al.*, 1984], au-delà force est de constater que l'on revient à des heuristiques plus ou moins heureuses du point de vue du classement [Lerman et Costa, 1996] [Chou, 1991], [Quinlan, 1993a] par exemple proposant un algorithme glouton de fusion deux à deux des sommets issus d'un éclatement jusqu'à optimisation du critère d'évaluation des segmentations. Bien entendu, il faut dans ce cas que la mesure soit adaptée, et pénalise les attributs multivalués.

Les graphes constituent une généralisation de ces pratiques de regroupements. Ils autorisent non seulement la fusion des sommets issus des modalités d'un attribut sur un noeud, mais également la fusion des sommets provenant de noeuds ayant une parenté plus ou moins lointaine (figure 7.3). Cette absence de contrainte permet de contourner les difficultés rencontrées par les premiers algorithmes de recherche de séquences de sous-arbres dans les arbres de décision [Mahoney et Mooney, 1991].

7.3 Nécessité d'une évaluation globale de la partition

A la différence des arbres, lors de la construction du graphe, nous aurons à comparer les gains respectifs d'une opération de fusion avec une opération de segmentation, donc comparer des partitions contenant un nombre différent de sous-groupes. La question que l'on se pose est : "Peut-on généraliser les mesures de qualité de segmentation à l'évaluation de la partition globale?".

Il est clair qu'en choisissant l'induction par graphes, nous intégrons une préférence pour les partitions plus concises que l'on peut matérialiser par un nombre de feuilles faibles par rapport à un arbre classique [Oliveira, 1994]. Il est primordial que la mesure que nous utilisons pour évaluer la partition ne soit pas biaisée en faveur des découpages fins. La propriété de fusion que nous exigions aux mesures de qualité de segmentation est cruciale ici.

Au final, nous pouvons adopter comme mesure de qualité de partition globale d'un échantillon l'ensemble des mesures qui répondent aux 5 propriétés issues des travaux de [Zighed, 1985], en particulier ceux que nous avons mis en exergue dans la pénalisation des découpages impliquant les attributs multivalués.

7.4 Construction de graphes d'induction avec contraintes

Les premiers travaux en la matière sont certainement ceux de [Chou, 1988] qui se proposait de construire un graphe orienté sans cycle nommé "treillis de décision" dont la structure était prédéfinie avant même la construction du classifieur. D'autres travaux sont venus par la suite, l'idée de base était de partir d'un classifieur déjà construit, sous forme d'arbre de décision par exemple, mais ça peut être également un graphe d'induction produit par les méthodes décrites dans la section suivante, que l'on réduira au mieux en procédant à des fusions et/ou à des suppressions de noeuds [Oliveira et Sangiovanni-Vincentelli, 1995].

Le point commun de ces algorithmes est de contraindre au préalable la structure du graphe, du point de vue de la fréquence et du lieu d'apparition des variables discriminantes. Afin de mieux saisir le fonctionnement d'un de ces algorithmes, celui de [Kohavi et Li, 1995], nous donnons ci-après une série de définitions tirés de [Kohavi, 1995b].

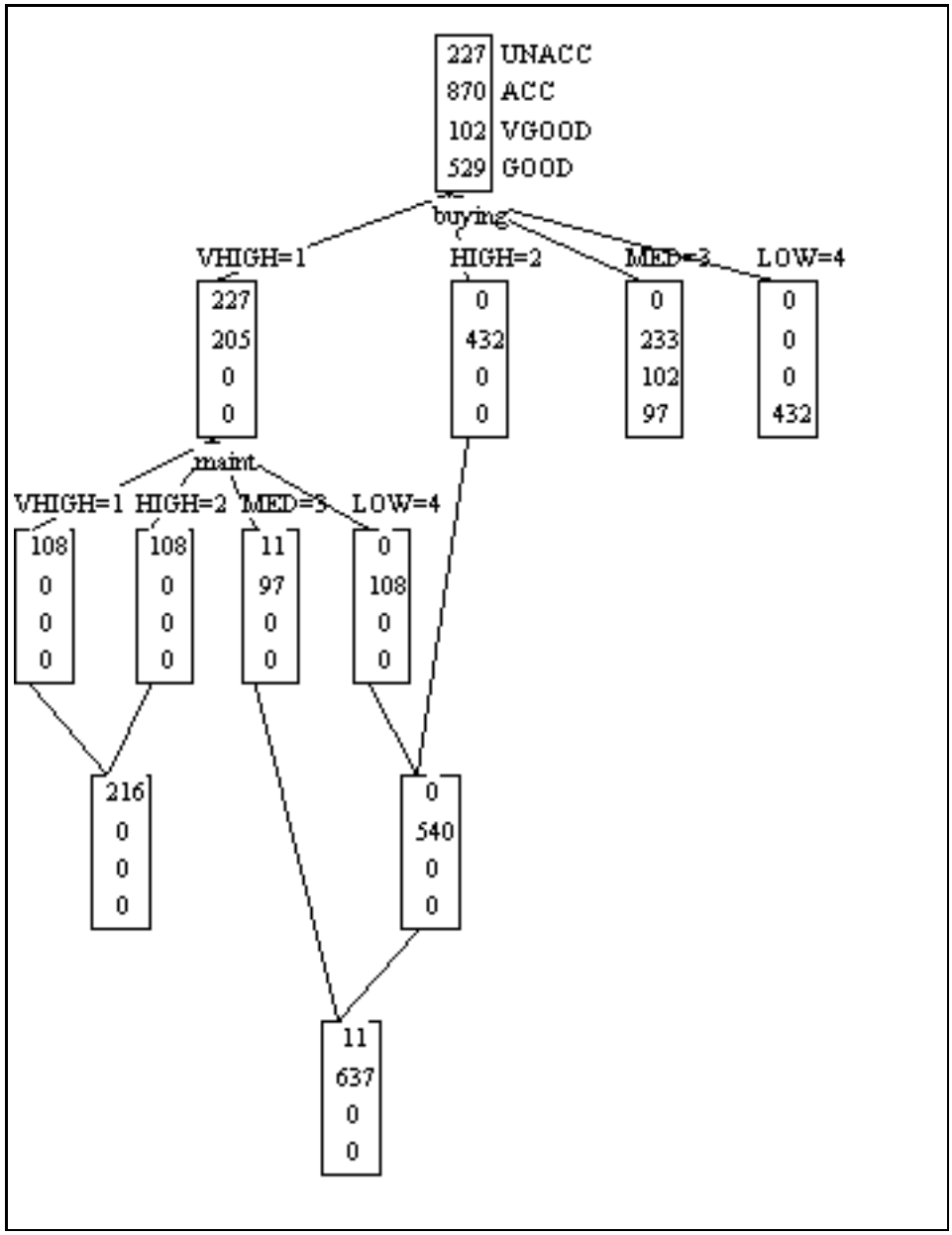

FIG. 7.3 – *Exemple de fusions entre sommets issus du même père et/ou d'ascendance lointaine*

7.4.1 Définitions et restrictions

Un graphe de décision est un graphe orienté sans cycle permettant de classifier une variable à prédire contenant K classes. Il a les propriétés suivantes :

- il possède exactement K feuilles[30], chacune d'elle correspondant à une classe;
- un noeud non-terminal s'appelle noeud de branchement[31], il possède un label qui correspond à la variable de segmentation et possède l arcs correspondant au nombre de modalité de cette variable;
- le premier noeud s'appelle la racine, il ne possède pas de père.

A cette première définition, on peut apposer des restrictions dans la structure des graphes que nous utilisons en induction [Oliveira, 1994][Kohavi, 1995b] :

- *graphe "read-once":* pour tous les chemins partant de la racine à une feuille, un attribut prédictif n'apparaît qu'une seule fois;
- *graphe à niveaux*[32] : les noeuds du graphe sont subdivisés en une séquence d'ensembles disjoints par paires de manière à ce que les arcs de sortie de tous les noeuds de même niveau aboutissent au niveau suivant;
- *graphe "oblivious"*: tous les noeuds d'un même niveau portent le même label, i.e sont segmentés avec la même variable;
- *graphe réduit*[33] : aucune segmentation sur deux noeuds distincts n'aboutit à la même subdivision, si c'est le cas, ils seront fusionnés.

A partir de tout ou partie de ces restrictions, la plupart des algorithmes connus essaient de dériver un graphe de décision réduit en partant d'un arbre [Chou, 1988] [Kohavi et Li, 1995] [Oliveira, 1994]. La principale limite ici étant le besoin d'une quantité suffisante d'observations pour recouvrer le concept à apprendre. Certes il existe d'autres méthodes qui construisent directement le graphe sous contraintes [Kohavi, 1994], mais elles s'avèrent peu efficaces en présence d'attributs non-pertinents, et de toute manière trop lentes.

Dans ce qui suit, nous présentons succinctement l'algorithme de [Kohavi et Li, 1995]. Nous l'avons choisi parce qu'il semble le plus souple et le plus "réaliste" dans le cadre de l'apprentissage

30. category node
31. branching node
32. Levelled
33. reduced

Numéro	Y	X_1	X_2	X_3
1	1	1	2	1
2	1	1	2	1
3	1	2	1	1
4	1	2	1	1
5	1	2	1	1
6	2	2	2	2
7	2	1	2	2
8	2	1	1	1
9	2	1	1	1
10	2	1	1	1

TAB. 7.1 – *Fichier exemple*

symbolique. La méthode de [Oliveira, 1994] par exemple, originaire du monde de la logique[34] [Bryant, 1986], est limitée aux attributs booléens et à un problème à deux classes.

7.4.2 Elaboration d'un graphe sous contrainte

L'algorithme construit un graphe de décision "read-once" et "oblivious" à partir d'un arbre de décision "oblivious" en fusionnant les noeuds qui se trouvent au même niveau. Il comporte trois étapes distinctes que nous détaillerons sur les données exemples de la table 7.1.

Arbre de décision "oblivious"

La première étape consiste à construire un arbre de décision "oblivious" i.e à chaque niveau de l'arbre, tous les noeuds sont segmentés à l'aide de la même variable. La sélection des variables de segmentation se fait à l'aide du très classique gain d'entropie informationnel qui, pour pénaliser les attributs multivalués, est divisé par $\log(L_X)$, où L_X est le nombre de modalités de la variable $X(.)$ impliquée. L'arbre est construit jusqu'à ce que tous les sommets terminaux soient purs, ou que l'on satisfasse une règle d'arrêt. Notons que de la complexité de cet arbre dépend la concision du graphe que l'on tirera dans la phase suivante.

Il y a une différence notable par rapport aux arbres d'induction classiques [Breiman *et al.*, 1984] [Quinlan, 1986b], il peut être avantageux de procéder à des découpages de noeuds purs. En effet, sur un même niveau, il n'y a aucune raison que tous les noeuds présentent la même pureté, le gain induit sur certains noeuds peut être suffisamment important pour que l'on accepte la partition

34. le terme consacré pour désigner les graphes de désicion dans ce domaine est "diagramme de décision binaire ordonné"

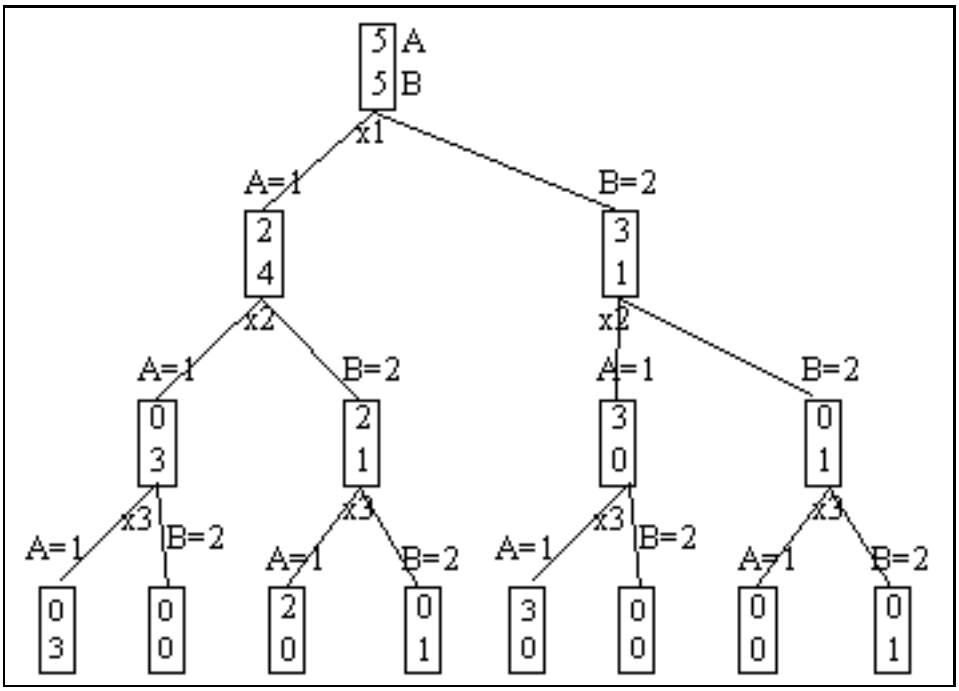

FIG. 7.4 – *Arbre oblivious sur les données de la table 7.1*

suivante. De toute manière, par les opérations de fusions et d'élagage de l'étape suivante, de tels découpages peu pertinents seront annihilés.

Pour notre exemple (table 7.1), nous observons l'arbre "oblivious" avec ses différents niveaux, toutes les feuilles sont pures.

Fusions de noeuds

L'étape suivante est la fusion de noeuds qui se trouvent sur le même niveau. On comprend maintenant pourquoi il était important que le premier arbre de décision soit "oblivious". Cette fusion doit obéir à certaines conditions.

Nous disons que deux sous-arbres d'un graphe sont **isomorphiques** si et seulement si

- à leurs feuilles sont assignées les mêmes classes,
- ou, les noeuds racines contiennent les mêmes tests, et que leurs noeuds enfants sont des racines de sous-arbres isomorphiques.

La fusion de deux sous-arbres isomorphiques ne change pas le comportement en classement du graphe de décision. Notons que les feuilles ne contenant aucun individu, auquel nous ne pouvons assigner de classes, reçoivent le label "inconnu". Puisque qu'implicitement, il a une préférence pour les graphes réduits, contenant peu de noeuds, l'auteur élargit la définition de sous-arbres

isomorphiques à sous-arbres compatibles où les feuilles de divergences sont uniquement des feuilles auxquelles sont assignées la valeur "inconnu".

Lorsque les données sont bruitées, la notion de compatibilité ci-dessus est trop restrictive, il se peut très bien qu'une feuille contienne un individu lui assignant une classe, alors qu'il ne s'agit en réalité que d'un bruit. L'auteur définit la k-compatibilité qui correspond à l'accroissement toléré de l'erreur lorsque l'on passe des sous-arbres isomorphiques au sous-arbre fusionné. Comment fixer la valeur k ? L'auteur adopte une démarche proche de celle de [Quinlan, 1993a] avec le taux d'erreur pessimiste. Soit v_a le nombre d'individus mal classés sur les deux sous-arbres isomorphiques, nous pouvons en déduire v_b la borne haute de l'intervalle de confiance pour un risque critique $1 - \alpha$ donné. Le nombre d'erreurs toléré pour le passage à la fusion sera donc $k \leq (v_b - v_a)$.

Lorsque v_a est égal à zéro, l'auteur prétend que l'on ne peut pas calculer v_b qui sera également nul. Nous remarquons que ceci est faux, il existe des méthodes exactes de calcul des intervalles de confiance lorsque les probabilités associées sont très faibles [Ventsel, 1973]. [Kohavi, 1995b] propose la cross-validation pour obtenir une estimation plus "honnête" de l'erreur. Il utilise la même structure d'arbre i.e les mêmes séquences de variables sont introduites. De fait, seules les assignations de classes sur les feuilles peuvent changer d'un essai sur l'autre, ce qui peut arriver très souvent lorsque les effectifs sont très faibles, l'erreur est estimée sur le reste de l'échantillon.

De l'arbre "oblivious" de la figure 7.4[35], nous extrayons le graphe (figure 7.5) en utilisant le principe de la fusion des sous-arbres isomorphiques.

Elagage par le bas[36]

Enfin la dernière étape consiste à supprimer les feuilles non-pertinentes en utilisant le principe de l'élagage. La stratégie est proche de celle de [Quinlan, 1993a] : on élague si l'augmentation de l'erreur induite est inférieure à k, où k est déterminé à l'aide de la procédure précédente.

Dans notre exemple, nous constatons que cette étape supplémentaire permet de supprimer les éclatements sur des noeuds purs (figure 7.6).

Pour être franc, nous sommes assez sceptiques quant à l'efficacité de cette famille de méthodes sur des problèmes d'induction réels. Les différentes expérimentations [Kohavi, 1994] [Oliveira et Sangiovanni-Vincentelli, 1995] [Kohavi et Li, 1995] montrent d'ailleurs que lorsqu'il s'agit d'approximer des fonctions booléennes complexes (parité, m-of-n, ...), cette famille de stratégies se comporte très bien. En revanche sur les bases de données benchmark [Murphy et Aha, 1995], issues pour la plupart d'études de cas effectivement réalisées, elles se distinguent très rare-

35. Afin de clarifier la lecture de l'arbre, rappelons qu'en dessous des sommets figurent la variable de segmentation, sur les branches la modalité prise par la dite variable (sous la forme dénomination = code). Sur la droite du sommet initial, nous avons inscrit les modalités prises par la variable à prédire.

36. Bottom-up Pruning

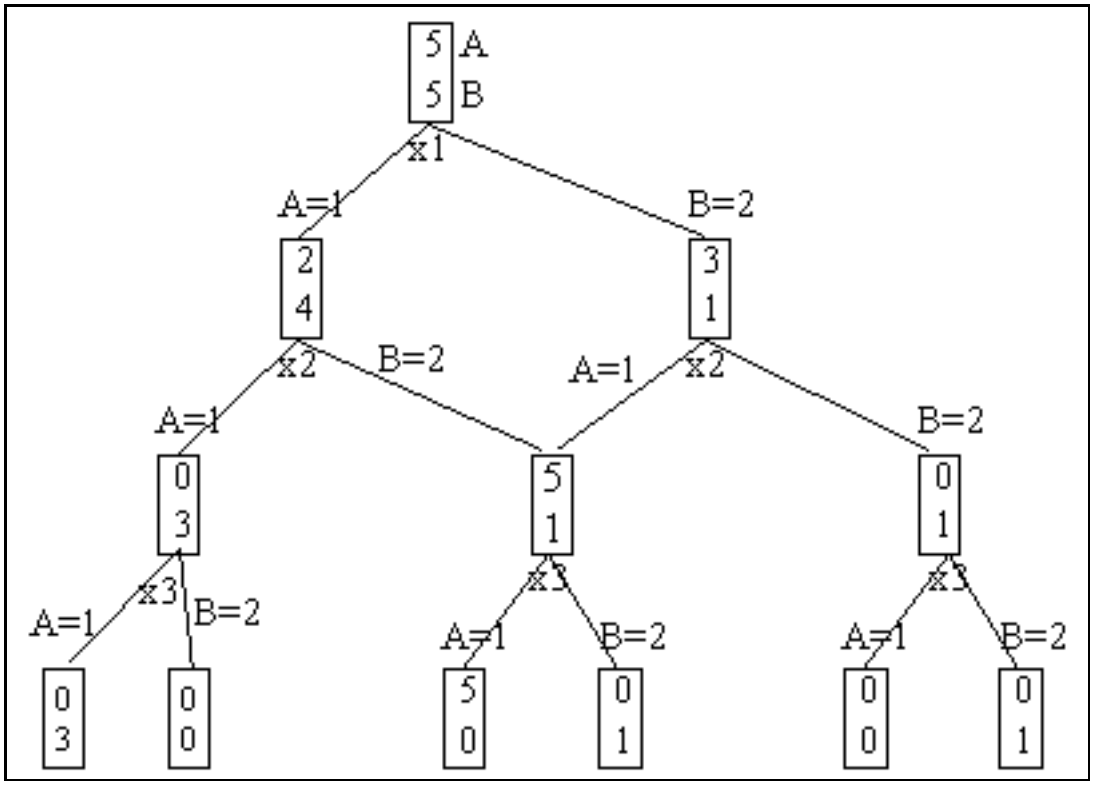

FIG. 7.5 – *Fusion de deux noeuds au deuxième niveau*

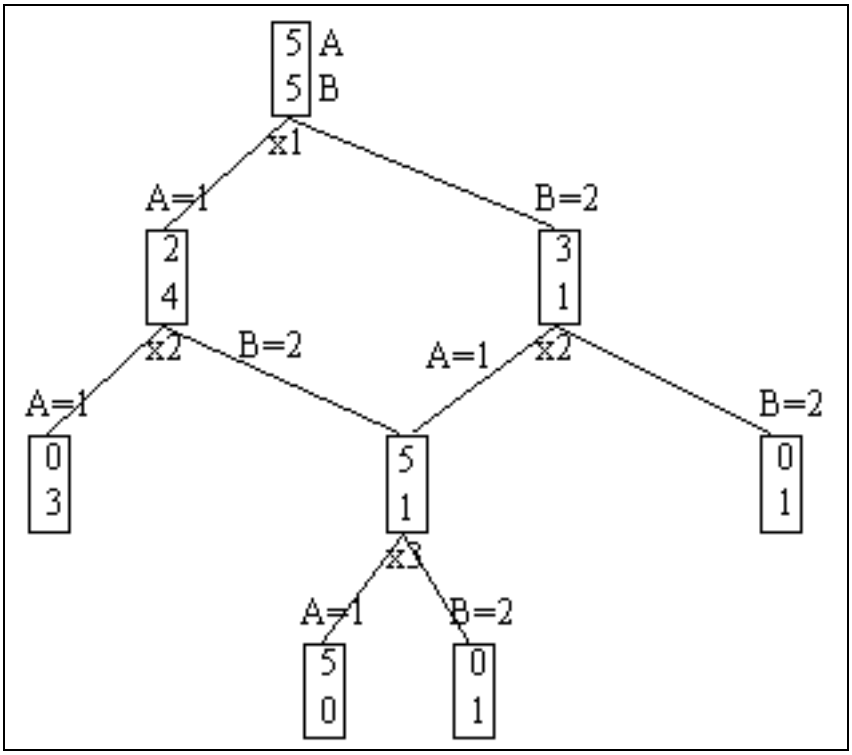

FIG. 7.6 – *Elagage pour retrouver une expression du graphe plus amène*

ment face à des stratégies éprouvées comme C4.5 [Quinlan, 1993a]. Cela semble naturel tant les contraintes de structures imposées à l'algorithme sont trop rigides et induisent vraisemblablement un biais trop élevé.

En fait, nous pensons qu'il s'agit là plutôt d'une piste très intéressante pour réduire sans perte les graphes d'induction construits sans hypothèses de structure que nous verrons plus bas. La suppression et la fusion de séquences de sous-graphes correspondrait à une phase d'élagage éliminant les portions non-pertinentes, améliorant ainsi les performances de l'algorithme glouton initial.

7.5 Construction de graphes d'induction hors contraintes

Contrairement aux méthodes de la section précédente, les algorithmes que nous décrivons ici recherchent itérativement la meilleure partition sans imposer une contrainte de structure : n'importe quelle variable peut intervenir à n'importe quel sommet dans la phase d'expansion du graphe, le nombre de sommets terminaux est quelconque. Il est même possible d'obtenir un arbre si cela induit l'optimisation de la mesure.

Historiquement, l'algorithme général de construction de graphes en absence de contraintes de structures a été élaboré par [Zighed, 1985]. Sans le savoir, [Oliver, 1993] en a repris les deux traits principaux que nous exposons ici :

1. une mesure globale d'évaluation : des partitions dans le premier cas, du graphe sous la forme d'une théorie dans le second cas;

2. un algorithme glouton d'exploration de la solution optimale.

La recherche de la solution optimale est NP-complet [Takenaga et Yajima, 1993], mais au contraire des arbres la construction des graphes introduit des nouvelles contraintes inconnues jusqu'alors.

7.5.1 Différence avec les algorithmes de construction des arbres

Ordonnancement de l'expansion du graphe

Dans les arbres de décision, l'expansion par la segmentation est récursive, les évaluations sur un sommet sont faites indépendamment des autres. Ce schéma n'est plus valable en ce qui concerne les graphes. En effet, à chaque étape de l'algorithme, nous comparons les mérites respectifs des éclatements et des fusions. L'état du graphe à l'étape t dépend de l'opération précédente à l'étape $t-1$.

Afin de mieux appréhender l'importance de cette distinction, nous allons prendre notre fichier exemple de 10 individus (Table 7.1). Après un premier éclatement impliquant la variable X1, nous

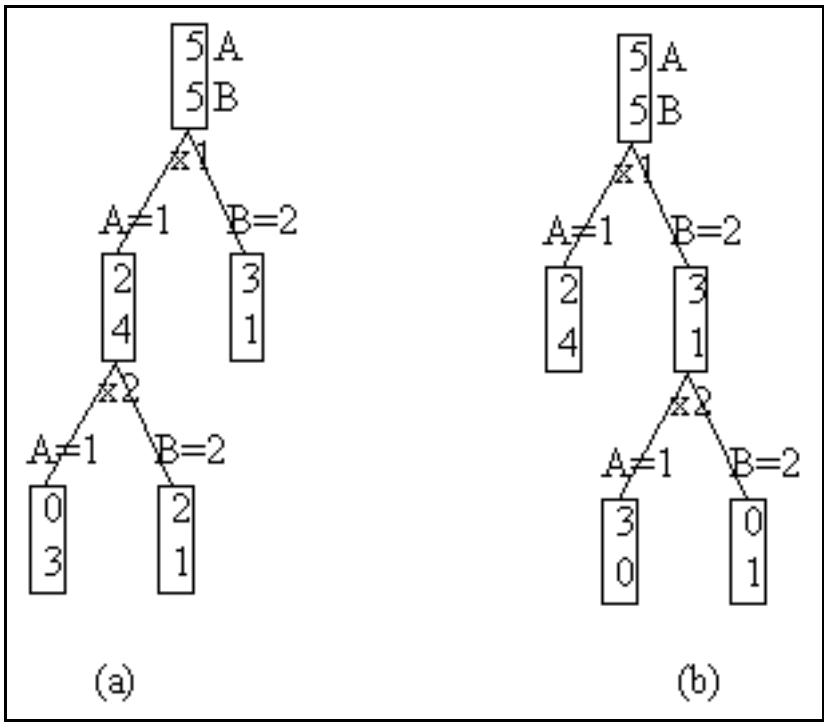

FIG. 7.7 – *Le choix de l'ordre de segmentation des sommets ne se pose pas dans la construction d'un arbre, ici il est crucial pour l'aspect final du graphe*

obtenons un graphe à deux feuilles. La fusion n'est pas pertinente à cette étape puisqu'elle nous ramènerait vers la partition initiale. Nous avons le choix entre deux segmentations. La première, à gauche, induit un gain[37] de 0.0688 avec la variable X2 [figure 7.7.a]. La seconde, à droite, toujours avec X2 obtient un gain de 0.0521 [fig. 7.7.b].

Dans la construction d'un arbre, la question de savoir s'il faut d'abord introduire la segmentation à gauche ou à droite ne se pose pas : le classifieur final n'est pas influencé par l'ordre de segmentation des sommets. Pour ce qui est des graphes, cette question est de première importance. En effet, si nous laissons se dérouler de manière automatique l'algorithme à partir de la situation de la figure 7.7.a (resp. 7.7.b), les graphes terminaux sont complètement différents : figure 7.8.a (resp. 7.8.b).

L'impossible élagage

Il est d'usage dans la construction des arbres de décision de produire la partition la plus fine que l'on réduira en évinçant les feuilles non pertinentes. Connue sous l'appellation d'élagage, on peut se demander si cette procédure est transposable dans les graphes.

Une des conditions d'élaboration de l'arbre maximum est d'accepter des partitions même si

37. Nous utilisons la mesure de gain d'entropie quadratique utilisant une estimation bayesienne des probabilités pénalisant les sommets à petits effectifs [Zighed et Rakotomalala, 1996a].

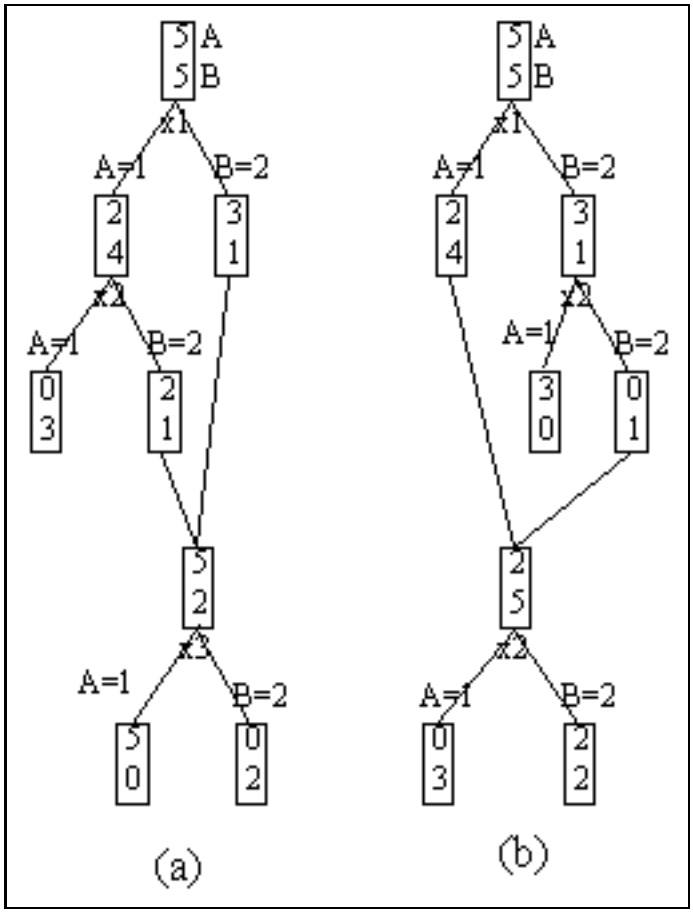

FIG. 7.8 – *Graphes résultants de choix différents au deuxième niveau*

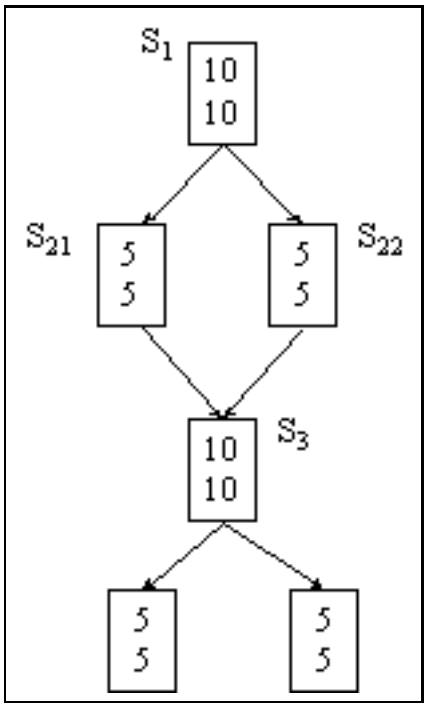

FIG. 7.9 – *Répetitions d'éclatement-fusion dans une construction "hurdling"*

elles ne sont pas pertinentes (construction "hurdling"), i.e n'améliorent pas la mesure d'évaluation globale de la partition. La condition d'arrêt naturelle devient alors la pureté des feuilles qui arrive assez vite puisque l'on fragmente énormément les données. Tout ceci devient inopérant lorsque l'on construit les graphes, il n'y a pas de fragmentation des données, l'expansion peut aller à l'infini d'autant plus que l'on s'expose à la succession d'opérations de segmentation et de fusion sur les mêmes sommets : c'est l'effet "guirlande". Dans l'exemple de la figure 7.9, nous voyons que la segmentation du sommet S_1 produit deux sommets S_{21} et S_{22} sans que cela soit pertinent. L'étape suivante vient avec la fusion, cette fois-ci pertinente, des sommets S_{11} et S_{21} en un sommet S_3, et ainsi de suite....

7.5.2 Algorithmes de construction des graphes

Des remarques de la section précédente découlent assez naturellement l'algorithme glouton de base de la construction d'un graphe [Oliver et Wallace, 1991][Zighed et Rakotomalala, 1996a] (table 7.2).

La procédure $Minimise\_Fusion()$ cherche la meilleure fusion de sommets deux à deux parmi toutes les feuilles composant le graphe, $Minimise\_Eclatement()$ cherche de manière triviale la segmentation qui induit le meilleur partitionnement.

$i = 0$
$T^i = (T_1)$
Repeter
  $i = i + 1$
  $T^i = T^{i-1}$
  $T^a = Minimise\_Fusion(T^i)$
  $T^b = Minimise\_Eclatement(T^i)$
  Si $\varphi(Ta) \leq \varphi(Tb)$
  Alors $T^{i+1} = T^a$
  Sinon $T^{i+1} = T^b$
  FinSi
Jusqu'à $\varphi(T_{i+1}) > \varphi(T_i)$
$T^* = T^i$

TAB. 7.2 – *Algorithme glouton d'élaboration des graphes d'induction*

Nous remarquons que malgré les doutes émis par [Kohavi et Li, 1995] sur la rapidité de l'algorithme, nous obtenons d'assez bons résultats dans la pratique. Certes, la recherche de la meilleure fusion est de complexité quadratique puisque nous faisons $\frac{L(L-1)}{2}$ tests[38], mais à la différence de l'éclatement où nous avons besoin de connaître la liste des individus qui y sont présents pour calculer les fréquences empiriques dans les sommets enfants, la distribution des classes peut être formée par simple addition des quantités n_{kl} stockées sur les sommets impliqués dans la fusion. De toute manière, ces calculs supplémentaires sont d'autant moins pénalisants que les graphes comportent, comparativement aux arbres, peu de sommets terminaux.

Face à cet algorithme élémentaire, il existe plusieurs raffinements qui ont été introduits par [Zighed *et al.*, 1992] :

- *préférence pour la fusion* : particulièrement lorsque les effectifs sont faibles, on peut avoir intérêt à fusionner dès que cette opération est justifiée i.e $\varphi(T^a) \leq \varphi(T^i)$. Ainsi, nous assurons à tous les sommets du graphe une consistance qui d'une part assure la fiabilité de la règle qui lui est associée s'il reste au final une feuille, et qui d'autre part, donne de meilleure chance à l'induction de trouver des segmentations plus sûres en diminuant la variance de l'indicateur de qualité de partition calculé.

- *opération de fusion-segmentation* : elle est analogue à la recherche en avant dans les algorithmes de construction d'arbres classiques. Elle consiste à rajouter une étape supplémentaire dans la recherche de la fusion en testant sur chaque sommet regroupé la segmentation

38. L est le nombre de feuilles dans le graphe

la plus pertinente, cela permet de mettre en évidence une troisième partition T^c que l'on confrontera aux autres partitions T^a et T^b.

Au final, nos différentes expérimentations ont montré que ces variantes n'apportaient d'améliorations significatives ni sur le classement en généralisation, ni sur la complexité du graphe.

7.6 Expérimentation

Nous voulons dans cette expérimentation vérifier si l'avantage théorique des graphes sur les arbres est confirmé par l'induction sur données réelles, i.e sa capacité à mieux apprendre et sa frugalité en échantillon d'apprentissage. Pour ce faire, nous avons procédé en deux phases :

- répétition 10 fois du partage aléatoire du fichier en deux fractions, 60% pour l'apprentissage et 40% pour la validation. Les résultats, taux de succès en validation, sont consignés dans le tableau 7.3, les écarts significatifs sont signalés.

- fort de ce premier résultat, nous allons vérifier si l'avantage éventuel des graphes se confirme lorsque la taille relative de l'échantillon d'apprentissage diminue, nous sommes descendus à 30% (table 7.4).

Pour représenter les graphes, nous utilisons la méthode SIPINA [Zighed *et al.*, 1992]. Afin de mettre les deux stratégies (Arbres et Graphes) sur le même pied d'égalité, nous décidons d'utiliser le moteur SIPINA en utilisant uniquement la segmentation. A l'instar du graphe, un gain négatif fait office de règle d'arrêt. Tous les autres paramètres de l'induction sont identiques (λ, α, taille minimale des sommets...).

Nous n'avons pas voulu comparer la complexité des classifieurs produits par les deux méthodes. En effet, il paraît difficile de mettre sur un pied d'égalité deux représentations de la connaissance différentes. Il est clair que les graphes auront toujours moins de sommets terminaux, les règles ne sont pas directement comparables puisque dans les graphes elles peuvent contenir des disjonctions, ce qui n'est pas le cas des arbres. Certes il eût été possible de quantifier la complexité des graphes en comptabilisant le nombre de bits nécessaires pour les décrire, nous n'avons pas adopté cette méthode parce que les techniques connues jusqu'à présent restent encore l'objet de débats passionnés, et sont loin de faire l'unanimité [Quinlan et Rivest, 1989] [Wallace et Patrick, 1993]. Notre seul indicateur sera donc le taux de succès en généralisation qui lui est complètement objectif.

Les résultats amènent quelques réflexions extrêmement intéressantes :

- la réduction drastique du fichier d'apprentissage (passage de 60% de la base à 30%) ne semble guère perturber les méthodes d'induction mis en oeuvre dans cette expérimentation.

Base	Arbre (1)	Graphe (2)	Ecart
autos	0.525±0.063	0.517±0.039	
breast	0.928±0.014	0.940±0.007	(1)<(2) **
car	0.875±0.008	0.875±0.008	
cpuperf	0.927±0.041	0.927±0.041	
credit	0.899±0.012	0.892±0.016	
flags	0.478±0.059	0.467±0.059	
hepatitis	0.783±0.057	0.797±0.059	
ionosphere	0.872±0.028	0.873±0.031	
iris	0.940±0.017	0.940±0.017	
lung	0.546±0.191	0.546±0.191	
pima	0.722±0.014	0.739±0.012	
vote	0.867±0.042	0.867±0.042	
wave	0.701±0.048	0.721±0.037	(1)<(2) *
wine	0.842±0.043	0.863±0.032	(1)<(2) *
zoo	0.795±0.112	0.791±0.092	

TAB. 7.3 – *Taux de succès en validation, 40 % de l'échantillon*

Base	Arbre (1)	Graphe (2)	Ecart
autos	0.492±0.063	0.480±0.050	
breast	0.926±0.157	0.927±0.016	
car	0.877±0.003	0.877±0.003	
cpuperf	0.849±0.017	0.869±0.049	
credit	0.873±0.028	0.873±0.033	
flags	0.432±0.056	0.432±0.056	
hepatitis	0.793±0.032	0.795±0.034	
ionosphere	0.876±0.027	0.880±0.028	
iris	0.927±0.045	0.927±0.045	
lung	0.361±0.064	0.361±0.064	
pima	0.721±0.176	0.727±0.026	
vote	0.830±0.069	0.830±0.069	
wave	0.661±0.051	0.675±0.043	
wine	0.855±0.057	0.855±0.057	
zoo	0.672±0.186	0.672±0.186	

TAB. 7.4 – *Taux de succès en validation, 70 % de l'échantillon*

Mis à part les fichiers (cpuperf, lung, wave, zoo), nous ne notons pas une baisse significative des taux de succès en validation, ce qui semble indiquer des concepts assez simples à apprendre, du moins sur lesquels des petits arbres marchent parfaitement.

- en observant attentivement les résultats, cette première impression est confirmée par le fait que dans le second fichier (Table 7.4). 8 fois sur 15 SIPINA produit des arbres parfaitement identiques à la stratégie "segmentation", ce rapport est de 5 fois sur 15 dans le premier fichier (Table 7.3).

- de fait, nous observons complètement le phénomène inverse de ce qui était attendu, sur les petits fichiers les graphes ne se démarquent pas du tout.

- pourtant, dans la première série d'expérimentation, ils s'imposent, de peu certes, sur trois fichiers.

Cet étrange bilan ne doit pas nous induire en erreur, l'avantage des graphes est indéniable, mais uniquement dans certains cas précis. En effet, toutes les expérimentations sur données synthétiques montrent que les concepts prenant la forme de fonctions booléennes disjonctives sont mieux appris par les graphes qui améliorent spectaculairement les performances des arbres, surtout lorsque la taille de la base d'apprentissage est assez faible [Oliver, 1993] [Oliveira, 1994] [Kohavi et Li, 1995]. En revanche, dans les mêmes travaux, on constate que sur des bases réelles, cette prééminence disparaît.

Il existe une explication simple à cette déconvenue : nous devons garder à l'esprit que le classifieur n'est qu'une émanation d'un système de représentation que l'on s'est choisi arbitrairement. Sur une base de données réelle, le concept sous-jacent de désignation des classes peut suivre un tout autre processus, que l'on ne peut exprimer soit parce que toutes les variables ne sont pas disponibles, soit parce que le concept prend une forme trop complexe. Dans les fichiers de notre expérimentation, les formes disjonctives ne semblent pas évidentes, du moins ne sont pas discernables.

Plus étrange a priori est le progrès des graphes sur les fichiers (breast, wave, wine) lorsque la taille de la base d'apprentissage augmente. A notre avis, il s'agit là d'une émanation de la meilleure résistance à la fragmentation. En effet, avec le surplus d'individus arrivent également plus de bruits. Les arbres ont tendance à "coller" aux données, d'autant plus que la méthode que nous avons utilisée ne dispose pas de dispositifs sophistiqués (règle d'arrêt par test statistique, élagage...). Regrouper des sommets de distributions proches permet de ne pas tomber dans le piège des règles de décision mal assurées car portées par un nombre d'observations trop faible [Zighed *et al.*, 1992]. Notons néanmoins que ce phénomène n'est pas généralisé, sur les autres fichiers les deux stratégies se valent. Peut-être faut-il y voir une interaction avec le processus de

discrétisation? Dans les trois fichiers sus-cités, tous les attributs prédictifs sont continus. Nous avons pourtant dans les deux cas (arbres et graphes) utilisé la même discrétisation binaire locale.

7.7 Conclusion

Les graphes d'induction répondent de manière théorique à certaines insuffisances des arbres, notamment la réplication des sous-arbres et la fragmentation des données. Sur des données artificielles contenant des formes disjonctives, ils se démarquent significativement dans les études empiriques. Le fait que cet écart ne soit pas manifeste dans ce chapitre s'explique aisément par la nature des données disponibles sur le serveur UCI Irvine où la plupart les arbres de petite taille sont suffisants pour décrire les concepts.

Notons que sur le conseil d'un de nos examinateurs, nous avons systématisé l'étude précédente en augmentant d'un pas de 20% la taille du fichier d'apprentissage. Les résultats ne sont pas probants, on ne voit pas se dessiner une tendance réelle dans le comportement des graphes face aux arbres. Ceci tient d'une part aux spécificités des fichiers issus de la base Irvine, d'autre part, il était difficile dans le cadre de cette étude de porter un jugement tranché du fait de l'impossibilité de construire un test statistique pour caractériser une évolution. En effet, on retrouve des fractions d'individus identiques dans les différents fichiers de validation qui ne sont pas de même taille, un traitement apparié étant exclu, l'absence d'hypothèse d'indépendance ne permet pas de spécifier un test de comparaison.

En tous les cas, les graphes constituent une alternative globale et élégante face aux méthodes plus complexes de constitution de variables intermédiaires afin d'augmenter leur pouvoir de représentation des arbres.

Chapitre 8

Construction de variables synthétiques

8.1 Introduction

Une des principales causes de l'échec lors de l'apprentissage est l'inadéquation de l'espace de description des individus par rapport au système de représentation des connaissances que l'on utilise. Il en résulte généralement, du moins dans les graphes d'induction, une fragmentation exagérée de l'ensemble d'apprentissage accompagnée d'un modèle sur-dimensionné, notamment avec la réplication des sous-arbres que nous avons vue dans le chapitre précédent. En enrichissant le système de représentation, les graphes, par rapport aux arbres, ont permis de pallier en partie ces faiblesses, mais il est clair que les attributs issus d'un système de bases de données, optimisées à des fins propres aux systèmes d'informations, ne sont pas nécessairement adaptés pour l'apprentissage [Holsheimer et Siebes, 1994].

L'élaboration de variables synthétiques[39] est une création algorithmique de nouveaux attributs descriptifs à partir des attributs de l'espace de représentation originel, et éventuellement formés à l'aide des connaissances du domaine. Elle possède la propriété d'invariance de classe i.e chaque individu garde sa classe d'appartenance quel que soit le nouvel espace de représentation construit. Sa conjonction avec l'induction reçoit le nom d'*induction constructive* [Michalski, 1983]. Elle est censée permettre une compression de la représentation du concept à apprendre et induire une meilleure précision au classifieur [Pagallo et Haussler, 1990] [Matheus, 1990].

Généralement, on pense que le passage d'attributs de bas niveau à des concepts intermédiaires est par nature intimement lié au domaine d'étude. En effet, le nombre de combinaisons possibles des variables en grand nombre est exponentiel. Par exemple, pour connaître les échelles de prix d'appartements à louer à partir de leur longueur et de leur largeur, il suffit de construire leur produit qui équivaut à la variable "surface". Dans une base de données contenant une centaine d'attributs, il faudrait des années pour que la machine trouve ce nouvel attribut en testant toutes

39. Feature construction

les combinaisons arithmétiques possibles, si l'on se restreint uniquement à des attributs prédictifs continus. De fait, un grand nombre de travaux sont centrés sur l'élaboration de nouvelles variables à partir d'experts [Towell *et al.*, 1990] [Rendell et Seshu, 1990].

Doit-on pour autant abandonner l'idée d'une construction automatique de variables intermédiaires ? Non, car en se limitant à des familles de combinaisons, en rapport avec les faiblesses présumées de la méthode d'induction mise en oeuvre pour l'apprentissage, nous pouvons espérer des améliorations de l'espace de représentation qui rejailliront sur la qualité et la concision du classifieur construit, mais aussi donneront des indications aux experts sur les concepts récurrents qui demandent des interprétations plus soignées.

Les premiers travaux sur la construction automatique de variables de haut niveau dans l'élaboration de graphes d'induction ont certainement été ceux de [Henrichon et Fu, 1969]. Dans ce chapitre, nous nous concentrerons sur plusieurs problèmes qui peuvent peser sur les performances des graphes d'induction, et nous décrirons les solutions afférentes[40] :

- la fragmentation des données : surtout lorsque l'on utilise des attributs prédictifs pouvant prendre de nombreuses valeurs (jusqu'à 20^4 dans certains cas [Lerman et Costa, 1996]), il est nécessaire d'effectuer des regroupements afin d'éviter l'apparition de nombreux sommets enfants avec très peu d'individus, ou carrément vides.

- la réplication des sous-arbres : notamment dans la construction des arbres, il arrive que certaines séquences soient répétitives, il peut être avantageux de combiner les attributs mis en cause pour réduire la taille de l'arbre et obtenir ainsi des feuilles plus consistantes [Matheus et Rendell, 1989].

- l'interaction entre variables : avec une construction gloutonne dans laquelle les variables sont testées puis introduites unes à unes, les graphes d'induction sont incapables d'appréhender les concepts où la description de la classe n'est perceptible qu'avec la projection des individus sur un espace à plusieurs dimensions. L'exemple le plus célèbre est la fonction XOR. Il faut alors mettre au point des stratégies de recherche en avant qui ne soient pas gourmandes en temps de calcul [Buntine, 1991].

- cas particulier des attributs continus : les graphes n'utilisent que les distributions conditionnelles des classes pour juger de la similarité des exemples. Sachant qu'ils sont plongés dans un espace euclidien il est certainement très intéressant de tenir compte de la proximité des individus lors de la construction du classifieur [Oliver et Dowe, 1995].

40. Certaines des solutions décrites ici ont d'ailleurs motivé le passage des arbres aux graphes. Nous les décrirons quand même parce que d'une part le passage aux graphes ne les ont pas pour autant rendues caduques, d'autre part elles contribuent à une meilleure compréhension du rôle de la construction automatique de variables dans l'induction.

Toutes les solutions associées aux problèmes ci-dessus trouvent leur aboutissement dans une interprétation cohérente des variables synthétiques construites. C'est rarement le cas dans la pratique, d'ailleurs il arrive très souvent que les règles issues de l'induction qui alimentent les systèmes à base de connaissances correspondent à des concepts nouveaux, inconnus des experts [Michie, 1979]. Il n'en reste pas moins que l'adjonction de ces nouvelles variables dans la base d'apprentissage permet la construction de graphes plus concis avec une meilleure capacité d'apprentissage traduite à travers une précision en classement meilleure et une diminution de la taille d'échantillon nécessaire pour apprendre les concepts.

L'organisation de ce chapitre sera directement inspirée des problèmes que nous avons soulevés ci-dessus. Dans un premier temps, nous nous intéresserons à la modification de la définition de variables par regroupement de ses modalités. Puis, nous aborderons deux axes de la construction de combinaisons booléennes de variables, le premier par construction itérative d'un arbre de décision, le second par recherche en avant à chaque sommet du graphe. Nous proposerons une problématique de la construction de variables de synthèse dans le cas continu, nous y testerons quelques solutions. Nous conclurons enfin.

8.2 Regroupement des valeurs d'un attribut

Le regroupement des modalités d'une variable semble nécessaire dès que l'on pense que certaines de ces modalités "vont ensemble" pour décrire le concept. On peut penser par exemple que dans certaines situations un attribut prédictif "lettre de l'alphabet", qui prend 26 modalités, peut être scindé en deux groupes (les voyelles et les consonnes). Le regroupement des modalités en deux groupes constitue alors le passage d'un attribut de bas niveau à un attribut de haut niveau [Berckman, 1995].

Ce passage pose plusieurs problèmes méthodologiques qu'il convient d'analyser en détail pour en évaluer l'effet dans l'induction :

- faut-il toujours construire une variable binaire [Cestnik *et al.*, 1987a], ou est-il parfois avantageux de procéder à un regroupement ramenant à une variable possédant plus de deux modalités [Cheng *et al.*, 1988]? Et dans ce cas, comment peut-on fixer le nombre de modalités de la nouvelle variable?

- quel critère optimiser lors de cette phase?

- existe-t-il un algorithme optimal de transformation travaillant dans un temps raisonnable [Breiman *et al.*, 1984], dans le cas contraire quel est l'effet de l'adoption des heuristiques de recherche [Berckman, 1995]?

A travers la présentation des différentes options de regroupement adoptées dans la littérature des graphes d'induction, nous essayons de répondre à ces questions en mettant en avant soit des réponses empiriques issues d'expérimentation, soit des avantages que l'on justifie intuitivement. Mais auparavant, essayons de recenser les avantages "théoriques" du regroupement des modalités d'une variable.

8.2.1 Intérêt du regroupement des modalités d'une variable

Le premier avantage que l'on a à regrouper certaines modalités d'une variable est certainement dans la lutte contre la fragmentation des données. Lorsque l'attribut prédictif possède de nombreuses modalités, il arrive souvent que certains sommets enfants possèdent très peu ou pas du tout d'exemples. Ce dernier cas est très ennuyeux car la plupart des logiciels d'inductions par graphes possèdent souvent une règle de partitionnement qui refuse une segmentation si un ou plusieurs des sommets construits ont une taille inférieure à une valeur fixée à l'avance [Quinlan, 1993a][Zighed *et al.*, 1992]. Dès lors le regroupement de certaines modalités permet de lever en partie cette contrainte et de solutionner le problème des sommets enfants exempts d'observations. Dans l'exemple des lettres de l'alphabet, si nous ne disposons que de 20 individus, un partitionnement à l'aide de cette variable induit des sommets enfants avec une taille de 0.77 en moyenne. Certains sommets enfants sont vides. Si nous passons à l'attribut binaire "consonnes-voyelles", la taille moyenne des feuilles passe à 10, assurant ainsi une segmentation plus fiable.

Le second avantage toujours en faveur du regroupement est la concentration des modalités non-informatives de la variable. Cette idée est à l'origine de l'algorithme GID3 [Cheng *et al.*, 1988] que nous détailleront plus bas. Partant du principe que les sommets issus de la segmentation sont de qualités très inégales, et que la mesure indique une amélioration moyenne, on peut passer sans dommages à une variable annexe dans laquelle les modalités informatives sont mises en évidence et les autres seront regroupées dans une valeur "autres". Nous détaillerons cette méthode plus loin car elle constitue une alternative à la recherche de la partition optimale en m modalités.

Dans le cas particulier de la binarisation des variables [Breiman *et al.*, 1984], cette transformation permet de lever la difficulté de comparer les gains d'information induits par des attributs ne possédant pas le même nombre de modalités. Notons que cet avantage semble anecdotique maintenant que nous avons mis en évidence plusieurs mesures non-biaisées en faveur des attributs multivalués.

Enfin, dernier des avantages liées au regroupement des modalités d'une variable est l'élimination de la réplication des sous-arbres que nous avons introduits dans le chapitre sur les graphes d'induction. Mis à part dans [Berckman, 1995], cette solution est très peu citée dans la littérature où l'on préfère discuter de solutions introduites par création de combinaisons de variables

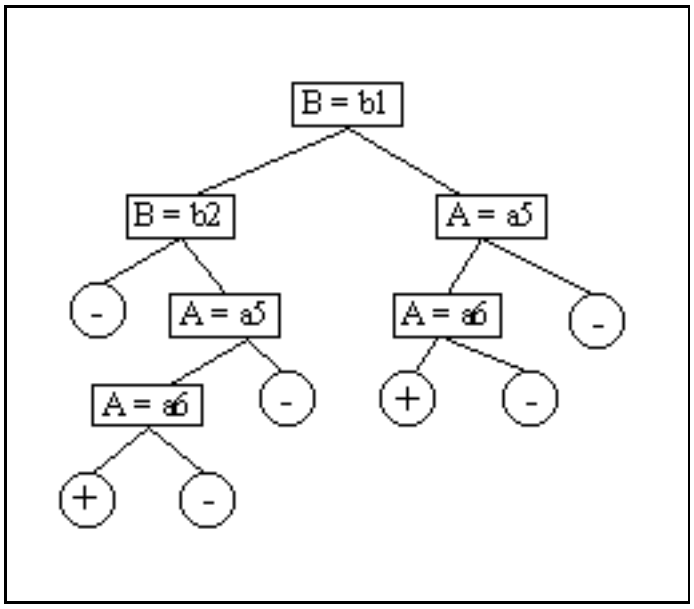

FIG. 8.1 – *Arbre minimum booléen*

ou le passage des arbres aux graphes. Pourtant, dans certaines situations, la simple binarisation permet d'y remédier sans que cela soit coûteux en terme de calcul. Reprenons l'exemple de [Berckman, 1995] afin d'illustrer nos propos. Soit deux attributs prédictifs A et B prenant respectivement leurs valeurs dans $\{a1, a2, a3, a4, a5, a6\}$ et $\{b1, b2, b3, b4, b5, b6\}$. La variable à prédire comporte deux classes $\{+, -\}$. Le concept à apprendre correspond à une fonction booléenne de la forme

$$f_+ = (a1 + a2 + a3 + a4).(b1 + b2) \tag{8.1}$$

Si nous utilisons l'algorithme de base d'ID3 où nous dérivons un sommet enfant à chaque modalité de l'attribut prédictif, le plus petit arbre consiste à segmenter le sommet initial avec la variable B, puis les sommets enfants correspondant aux modalités b1 et b2 avec la variable A. En tout, l'arbre contient 16 feuilles. Une première étape est de passer à un arbre booléen où nous introduisons le test $(A = al)$ sur chaque sommet, c'est déjà une première forme de binarisation des variables. Dans ce cas, l'arbre minimum contient 7 feuilles, avec des réplications de certaines portions de l'arbre (figure 8.1, par convention la branche droite correspond à la réponse $vrai$ et la gauche à la réponse $faux$). Si nous procédons maintenant au regroupement des modalités de A et B, nous obtenons un arbre à 3 feuilles (figure 8.2). Il est évident que de ces trois arbres, ce dernier a notre préférence parce que les règles de classement qui en sont issues sont nettement plus fiables puisqu'elles couvrent plus d'individus.

Dans ce qui suit, nous noterons $X(.)$ l'attribut prédictif, il prend ses valeurs dans $\aleph = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ avec $L = 6$.

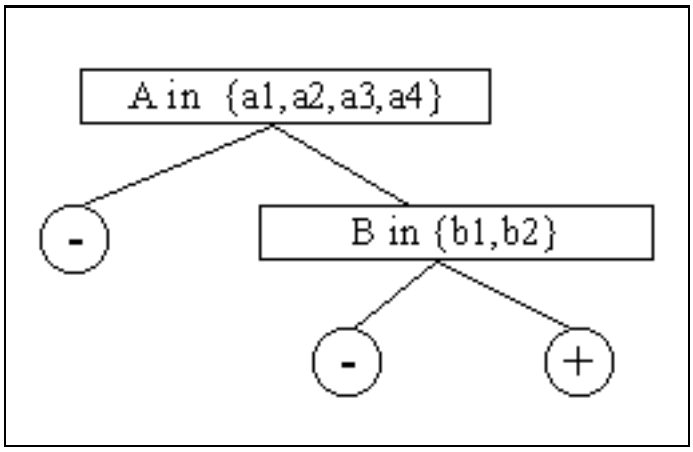

FIG. 8.2 – *Arbre minimum après regroupement des valeurs*

	x_1	x_2	x_3	x_4	x_5	x_6
y_1	9	5	2	8	4	2
y_2	1	5	8	2	6	8

TAB. 8.1 – *Tableau de contingence T^1 correspondant a la partition de l'echantillon d'apprentissge par un attribut prenant 6 modalites dans un probleme a deux classes*

8.2.2 La binarisation des attributs

La stratégie "une modalité contre les autres"

La manière la plus simple de binariser un attribut est de mettre en exergue une modalité particulière contre les autres, ce qui correspond en quelque sorte à un codage disjonctif complet de la variable. En effet en opposant deux groupes $\{x_l\}$ et $i \neq l \cup \{x_i\}$, cela revient à créer une variable booléenne $(X = x_l)$ qui prend la valeur "vrai" si la condition est vérifiée, "faux" sinon. Si nous notons $T(\{x_l\}|\{x_i/i \neq l\})$ le tableau de contingence correspondant à cette partition binaire, le choix de la segmentation de la variable $X(.)$ sur un sommet se fait en choisissant x_l tel que la mesure de qualité de segmentation $\Psi(T)$ soit optimale, où Ψ est n'importe quelle des mesures présentées dans le chapitre sur les indicateurs d'évaluation des segmentations. Le nombre de tests à effectuer est donc égal à L, le nombre de modalités de la variable $X(.)$.

Prenons un exemple simple pour illustrer notre propos. La partition de 60 individus par une variable $X(.)$ prenant 6 modalités dans un problème à deux classes a permis de dégager le tableau de contingence T^1 (Table 8.1) où nous calculons $\Psi(T^1) = 0.3981$.

En testant toutes les spécifications $T(\{x_l\}|\{x_i/i \neq l\})$, $l = 1, \ldots, L$, nous obtenons une série

de valeurs $\Psi(T)$ recensées dans la table suivante

	$\Psi(T)$
x_1	0.4531
x_2	0.5000
x_3	0.4736
x_4	0.4736
x_5	0.4971
x_6	0.4736

La spécification $T(\{x_1\}|\{x_2, x_3, x_4, x_5, x_6\})$ s'avère être la meilleure. Le découpage se fait en opposant la modalité x_1 aux autres, l'incertitude associée est égale à $\Psi(T) = 0.4531$.

Dans une éxpérimentation à grande échelle, [Munteanu, 1996] montre que ce type de spécification binaire est aussi bonne sinon meilleure, du point de vue des performances en classement, que la partition L-aire correspondant à la création d'un sommet enfant pour chaque modalité de l'attribut prédictif.

Regroupement optimal en deux modalités

Ce premier algorithme n'est finalement qu'un cas particulier de la recherche de la partition binaire optimale. Nous cherchons les deux sous-ensembles d'attributs $\aleph_g = \{x_a, x_b, \ldots\}$ et $\aleph_d = \aleph - \aleph_g$ tels que l'incertitude sur le tableau de contingence associé $\Psi(T)$ soit minimale, avec les contraintes suivantes : $card(\aleph_g) < card(\aleph)$, a prend ses valeurs dans $\{1, \ldots, 6\}$, b dans $\{1, \ldots, 6\} - \{a\}$, et ainsi de suite...

La manière la plus simple qui nous assure de trouver la partition binaire optimale est de tester tous les cas possibles, il y en a $2^{L-1} - 1$. Ceci est praticable si le nombre de modalités de la variable est faible, [Cestnik *et al.*, 1987a] le préconisent lorsque $L \leq 4$. Dans le cas contraire, il est évident qu'il nous faut trouver des heuristiques qui nous permettent de trouver cette partition en un temps raisonnable. Dans notre cas, il y a 31 cas possibles, si L=20 il y aurait 524.287 partitions à évaluer.

Certes, il existe plusieurs stratégies d'optimisation comme le recuit simulé ou les algorithmes génétiques, mais le plus rapide reste toujours l'algorithme glouton d'optimisation pas à pas. Nous montrons dans la table 8.2 le pseudo-code associé.

La procédure *introduire_meilleure_modalité*$(\aleph_g, \aleph_d)$ consiste à trouver la modalité contenue dans $\aleph_d$ tel qu'introduite dans $\aleph_g$ la partition induite minimise la quantité $\Psi(T_{\aleph_g}|T_{\aleph_d})$.

Les insuffisances de tels algorithmes sont connus, ils s'exposent aux dangers d'un optimum local. Mais dans la pratique [Berckman, 1995], ils semblent fournir de bons résultats : tant dans la recherche de la partition optimale que dans la qualité globale du graphe en tant que classifieur

$\aleph_g = \emptyset$
$\aleph_d = \aleph$
$\Psi^* = \infty$
Repeter
 introduire_meilleure_modalité$(\aleph_g, \aleph_d)$
 Si $(\Psi(T_{\aleph_g}|T_{\aleph_d}) < \Psi^*)$
 Alors
  Accepter modifications $\aleph_g, \aleph_d$
  $\Psi^* = \Psi(T_{\aleph_g}|T_{\aleph_d})$
 Sinon
  Rejeter modifications $\aleph_g, \aleph_d$
 FinSi
Jusqu'à $(\Psi(T_{\aleph_g}|T_{\aleph_d}) \geq \Psi^*)$ OU $(card(\aleph_d) = 1)$
$T = (T_{\aleph_g}|T_{\aleph_d})$ représente la partition optimale

TAB. 8.2 – *Algorithme glouton de recherche de la partition binaire optimale*

où l'on constate une réduction de la complexité de celui-ci sans diminution (ni amélioration d'ailleurs) des performances sur 11 bases benchmark. Cela ne nous étonne guère, nous avons déjà observé le même phénomène dans la discrétisation : l'optimalité en apprentissage est peu bénéfique face à un simple algorithme glouton.

Si nous reprenons notre exemple (Table 8.1), l'exploration peut être retracée à l'aide du tableau suivant. En gras sont indiqués les attributs extraits de $\aleph_d$ et introduits dans $\aleph_g$.

$\aleph_g$	$\{\}$	$\{x_3\}$	$\{x_3, x_4\}$	$\{x_3, x_4, x_5\}$
x_1	0.4531	0.4979	0.5000	0.4808
x_2	0.5000	0.4808	0.4688	0.4979
x_3	**0.4736**	$***$	$***$	$***$
x_4	0.4736	**0.4232**	$***$	$***$
x_5	0.4971	0.5000	**0.4043**	$***$
x_6	0.4736	0.4659	0.4980	0.4659

Au final, nous adoptons la partition binaire $\aleph_g = \{x_3, x_4, x_5\}$ et $\aleph_d = \{x_1, x_2, x_6\}$ avec $\Psi(T_{\aleph_g}|T_{\aleph_d}) = 0.4043$.

Cas particulier : la variable à prédire a deux classes

[Breiman *et al.*, 1984] dans le cas des problèmes à deux classes on proposé un algorithme optimal de détection de la partition binaire. Elle repose sur la convexité de la mesure d'éva-

luation de la segmentation et utilise des propriétés théoriques dûes à [Fischer, 1958]. Notons que [Asseraf, 1996] a utilisé le même principe pour rechercher la meilleure partition binaire en utilisant cette fois la mesure de Kolmogorov et Smirnov [Friedman, 1977].

Si l'on pose p_{1i} la proportion d'individus portant l'étiquette y_1 sachant qu'ils présentent la valeur x_i sur l'attribut $X(.)$. La partition binaire optimale est telle que

$$p_{1i} \leq p_{1j}, \forall x_i \in \aleph_g et x_j \in \aleph_d \tag{8.2}$$

Dès lors, la stratégie consiste à trier le tableau de contingence originel T en permutant les colonnes correspondant aux différentes modalités de $X(.)$ de manière à ce que la condition 8.2 soit respectée. La partition optimale est parmi les $L-1$ partitions adjacentes qui sont traduites dans le tableau T' trié.

Dans le cas du tableau 8.1, le tri permet de mettre en évidence T' qui prend la forme suivante

	x_3	x_6	x_5	x_2	x_4	x_1
y_1	2	2	4	5	8	9
y_2	8	8	6	5	2	1

Il ne nous reste plus qu'à trouver la segmentation binaire qui optimise $\Psi(T_{\aleph_g}|T_{\aleph_d})$ parmi

$\aleph_g$	$\Psi(T_{\aleph_g}\|T_{\aleph_d})$
$\{x_3\}$	0.4736
$\{x_3, x_6\}$	0.4232
$\{x_3, x_6, x_5\}$	0.4043
$\{x_3, x_6, x_5, x_2\}$	0.3955
$\{x_3, x_6, x_5, x_2, x_4\}$	0.4531

Nous trouvons ainsi la partition optimale qui est telle que $\aleph_g = \{x_3, x_6, x_5, x_2\}$ et $\aleph_d = \{x_4, x_6\}$ avec $\Psi(T_{\aleph_g}|T_{\aleph_d}) = 0.3955$. Nous constatons qu'il est différent de celui produit par l'algorithme glouton.

8.2.3 Généralisation : la m-arisation des attributs

La binarisation n'est pas toujours justifiée sémantiquement, on peut se demander si le regroupement d'un attribut à L modalités en m groupes est plus légitime dans certains cas. Prenons le cas d'une enquête d'option, il semble plus naturel que pour expliquer l'achat ou non d'une voiture, une variable retraçant la sensibilité d'un client au confort codé "très satisfait - satisfait - indifférent - mécontent - très mécontent" soit modifié en "satisfaits - indifférent - mécontents". Donc, en trois modalités.

Nous sommes rarement confrontés à ce cas de figure, où la partition et sa taille (en nombre de groupes) semblent évidentes. Dans la pratique, il nous faut décider quel est le nombre adéquat

de groupes, et comment les former. Par rapport à la discrétisation, la solution est nettement plus ardue à trouver du fait que l'on ne peut définir, sauf cas particulier des problèmes à deux classes, une relation de pré-ordre entre les modalités originelles de l'attribut prédictif.

Partitionnement m-aire par optimisation

Nous cherchons la partition en m groupes de l'attribut $X(.)$ qui comporte L modalités. Le nombre de cas possibles est donné par le nombre de Stirling du second ordre

$$Stirling\_2(L,m) = \frac{1}{L!}\sum_{i=0}^{m}(-1)^{L-i}\, C_m^i\, i^L$$

La comparaison de découpages en nombre de groupes différents ne pose pas de problèmes pour peu que nous adoptions une mesure d'évaluation des partitions non-biaisées en faveur des attributs multivalués. En revanche, il est clair que l'optimisation par l'exploration de toutes les partitions possibles est impraticable.

Certes, dans le cas où la variable endogène comporte deux classes, il nous semble possible d'étendre les résultats de [Breiman *et al.*, 1984] à l'optimisation en m groupes. Il suffirait alors de trier les modalités de $X(.)$ selon les probabilités p_{1i} est de mettre en oeuvre un algorithme de programmation dynamique analogue à celui que nous avons utilisé en discrétisation.

Dans le même ordre d'idées, toujours dans les problèmes à deux classes, en optimisant cette fois la mesure de gain d'entropie de Shannon, [Chou, 1988] a montré que la transformation optimale d'une partition en L groupes $(T_1|\ldots|T_L)$ en m groupes $(T_{\aleph_1}|\ldots|T_{\aleph_m})$ respecte la condition suivante

$$D(T_i||T_{\aleph_j}) \leq D(T_i||T_{\aleph_k}), \quad \forall j \in \{1,\ldots,m\},\ k \neq j,\ x_i \in \aleph_j$$

où $D(T_l||T_{l'})$ est la distance de Kullback-Liebler [Cover et Joy, 1991]

$$D(T_l||T_{l'}) = \sum_{k=1}^{K} p_{kl}.\log_2(\frac{p_{kl}}{p_{kl'}}) \tag{8.3}$$

De fait, la condition 8.3 réduit de manière drastique l'espace de recherche, et [Chou, 1988] a proposé un algorithme de complexité linéaire pour trouver la partition en m groupes. Notons que si l'auteur, faute de disposer d'une mesure adéquate, demande à l'utilisateur de spécifier la valeur m, l'utilisation de mesures capables de comparer des segmentations de taille différentes solutionne ce problème. Nous pourrions ainsi mettre en oeuvre la mesure d'association de [Theil, 1970] dérivée du gain d'entropie de Shannon.

Même si ces différentes approches donnent de bons résultats, ils restent sans effets dans un cadre général, avec un nombre quelconque de classes. Face à la difficulté de produire une méthode d'exploration satisfaisante, la seule solution envisageable est, à l'instar des travaux de [Breiman *et al.*, 1984], de constituer deux super-classes à partir de regroupements parmi les K classes.

Mise en exergue des modalités informatives

On peut également changer notre spécification du problème du regroupement. On peut par exemple ne vouloir distinguer que les modalités induisant une amélioration sensible de la mesure utilisée, les autres modalités étant alors regroupées dans une modalité "fourre-tout". Ceci constituerait alors une généralisation de la stratégie "une modalité contre les autres" au principe "les bonnes modalités contre les autres".

La méthode que nous présentons dans cette section est dûe à [Cheng *et al.*, 1988]. Les auteurs veulent avant tout construire un meilleur arbre en évitant les pièges de la sur-spécialisation. A cette fin, ils proposent de regrouper sous le label "autres" les modalités non-informatives de l'attribut, y compris celles pour lesquelles nous ne disposons pas d'observations.

L'algorithme est très simple : pour chaque attribut candidat à la segmentation, nous calculons l'incertitude E_l associée à chacun des sommets issus des différentes modalités. Nous sélectionnons le minimum

$$E^* = \min_{l=1,\ldots,L} E_l$$

et nous regroupons dans le label autres tous les sommets tels que

$$\aleph_{autres} = \{x_l \,/\, E_l > TL \times E^*\}$$

Lorsque $TL = \infty$, nous retrouvons l'algorithme de base ID3 [Quinlan, 1979] où pour chaque modalité de $X(.)$ nous créons une branche; $TL = 1$ correspond à la stratégie "une contre les autres". Dans la pratique, TL peut être optimisé par cross-validation.

Dans notre exemple (table 8.1), nous avons calculé l'incertitude associée à chaque sommet

	E_l
x_1	0.2778
x_2	0.5000
x_3	0.3750
x_4	0.3750
x_5	0.4861
x_6	0.3750

$E^* = 0.2778$, si nous fixons $TL = 1.4$: nous obtenons la partition suivante

	x_1	x_3	x_4	x_6	$\aleph_{autres}$
$Y = y_1$	9	2	8	2	9
$Y = y_2$	1	8	2	8	11

Testée sur des bases de données de l'industrie des semi-conducteurs, [Cheng *et al.*, 1988] ont montré empiriquement que leur méthode permettait de réduire la complexité des arbres

(en nombre de feuilles) et, ce qui est remarquable, améliorait le taux de classement d'ID3 en généralisation.

8.3 Construction itérative de combinaisons booléennes de variables

C'est l'étape au-dessus de la construction de variables synthétiques, elle met en jeu non plus une seule variable, mais deux ou plusieurs variables. Les travaux originels sont dûs à [Pagallo et Haussler, 1990][Matheus et Rendell, 1989], relayés par la suite par [Yang *et al.*, 1991b] et [Oliveira et Vincentelli, 1993].

8.3.1 Intérêt de la construction itérative de combinaisons booléennes de variables

L'idée de base est : "plus puissant sera le pouvoir de représentation des attributs prédictifs, plus simple en sera l'expression du classifieur construit" [Flann et Dietterich, 1986]. S'agissant des arbres de décision, on sait qu'ils ne sont pas adaptés pour décrire simplement les fonctions booléennes de forme disjonctive, ce qui entraîne la réplication des sous-arbres et la fragmentation des données. Dès lors, les séquences de noeuds situés sur les parties basses de l'arbre, même si elles sont pertinentes, seront éliminées par l'élagage tout simplement parce que les effectifs associés sont trop faibles. L'induction par arbres nécessite donc des échantillons de très grande taille.

Nous avons déjà discuté de ces insuffisances dans le chapitre consacré aux graphes d'induction. Nous y avons adopté la démarche inverse i.e augmenter le pouvoir de représentation du modèle. Ici, l'objectif serait plutôt d'augmenter le pouvoir prédictif des variables en constituant des combinaisons booléennes de variables.

Si nous reprenons l'exemple de la fonction booléenne $f = a.b + c.d$ (cf. chapitre ss, figure aa), l'arbre associé possédait 7 feuilles, le graphe 3. En construisant deux attributs supplémentaires $c_1 = a.b$ et $c_2 = c.d$, la fonction s'écrit $f = c_1 + c_2$. L'arbre correspondant contient alors 3 feuilles (figure 8.3). Nous retrouvons les performances des graphes, tout le problème est dans la stratégie adoptée dans la recherche de ces combinaisons.

8.3.2 Construction de combinaisons booléennes de variables par analyse topologique des arbres

Dans l'algorithme originel [Pagallo et Haussler, 1990] et les dérivés connus à ce jour [Oliveira et Vincentelli, 1993], on se restreint à un problème à deux classes $\{+, -\}$, tous les attributs sont booléens. Cette dernière limitation n'est pas préjudiciable, un codage disjonctif complet d'une variable qualitative quelconque permet d'y remédier. Les combinaisons construites

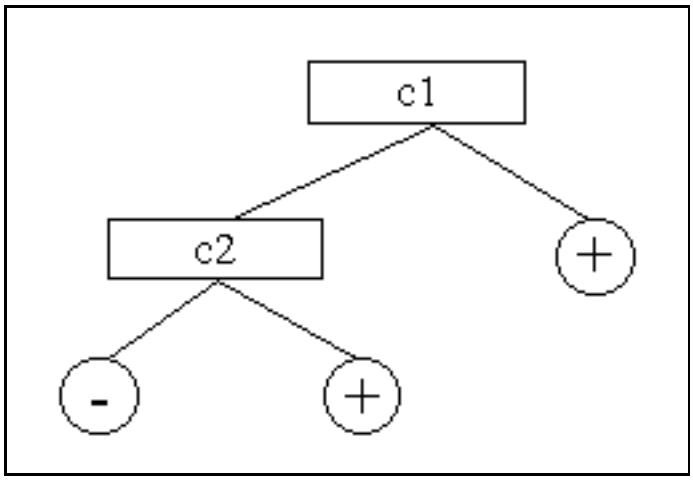

FIG. 8.3 – *Arbre après adjonction de nouvelles variables issues de combinaisons booléennes*

$k = 0$
$V_1 = V$
Repeter
$k = k + 1$
$T_k = Construire\_Arbre(\Omega^a, V_k)$
$F = Trouver\_Combinaisons(T_k)$
$V_{k+1} = V_k \cup F$
Jusqu'à $(Vk + 1 = Vk \quad ou \quad \|V_{k+1}\| \geq M)$
T_k est l'arbre final.

TAB. 8.3 – *Algorithme FRINGE*

sont sous la forme de conjonctions et de disjonctions de *propositions*, où *proposition* équivaut à un attribut ou à sa négation.

Dans ce qui suit, nous allons détailler l'algorithme d'extraction itératif des variables synthétiques, et les différentes formes extraites.

L'algorithme FRINGE

La méthode FRINGE [Pagallo et Haussler, 1990] est relativement simple : on construit un arbre de décision, on l'analyse pour déterminer les variables synthétiques candidates, qui seront par la suite introduites dans la liste des attributs prédictifs. On édifie à nouveau un arbre de décision, et ainsi de suite jusqu'à ce qu'aucun nouvel attribut ne soit construit. Le détail de l'algorithme est dans la table 8.3.

V est l'ensemble initial des attributs prédictifs, T_k est l'arbre construit avec l'échantillon Ω^a et l'ensemble des variables V_k, F est l'ensemble de nouvelles variables construites à l'itération k,

enfin la quantité M est fixée par l'utilisateur.

Le noeud principal de la stratégie se situe dans l'extraction des nouvelles variables dans l'arbre [la fonction $Trouver\_Combinaison(.)$]. Dans l'algorithme initial, les auteurs se sont contentés de construire de nouvelles variables constituées de conjonctions de propositions des deux derniers noeuds situés à la "frange" de l'arbre, i.e précédant les feuilles, menant à une conclusion positive.

Il y a plusieurs raisons au choix de cette stratégie :

- les variables sont construites par combinaisons de variables deux à deux;
- les noeuds situés sur la frange sont les moins sûrs puisqu'ils couvrent très peu d'individus de la base d'apprentissage;
- la répétition des séquences de sous-arbres a lieu le plus souvent dans la partie basse du modèle.

D'autres auteurs ont par la suite enrichi la liste des formes détectées avec le système FRINGE.

Liste des "formes" détectées

[Pagallo et Haussler, 1990] ont avant tout travaillé sur les conjonctions de propositions, où "propositions" avions-nous précisé correspondait à un attribut ou à sa négation. [Yang *et al.*, 1991b] y ont ajouté les disjonctions, et [Oliveira et Vincentelli, 1993] la forme XOR. En tout, nous recensons 12 formes représentées dans le graphique (réf. 8.4).

L'algorithme de base ne change pas, ces perfectionnements touchent uniquement au pouvoir de représentation des nouveaux attributs construits.

Limitations de FRINGE

Ce système a été avant tout conçu pour répondre à un problème théorique : comment représenter des fonctions booléennes en forme normale disjonctive[41] à l'aide d'arbres de décision. Dans la pratique, utilisé pour l'induction, on voit apparaître plusieurs inconvénients :

- FRINGE travaille essentiellement dans un monde booléen. Si cela ne pose guère problème pour les attributs prédictifs que l'on peut transformer aisément grâce à un codage disjonctif complet, il en est autrement de la variable à prédire. En effet les formes détectées sont avant tout associées à la conclusion positive $\{+\}$, ce système devient caduque dès que l'on sort du schéma exemples - contre-exemples, i.e du problème à deux classes.
- dans les bases de données réelles, nous sommes souvent confrontés au bruit. On présume généralement que ce phénomène se manifeste surtout dans les parties basses de l'arbre,

41. DNF : Disjunctive normal form

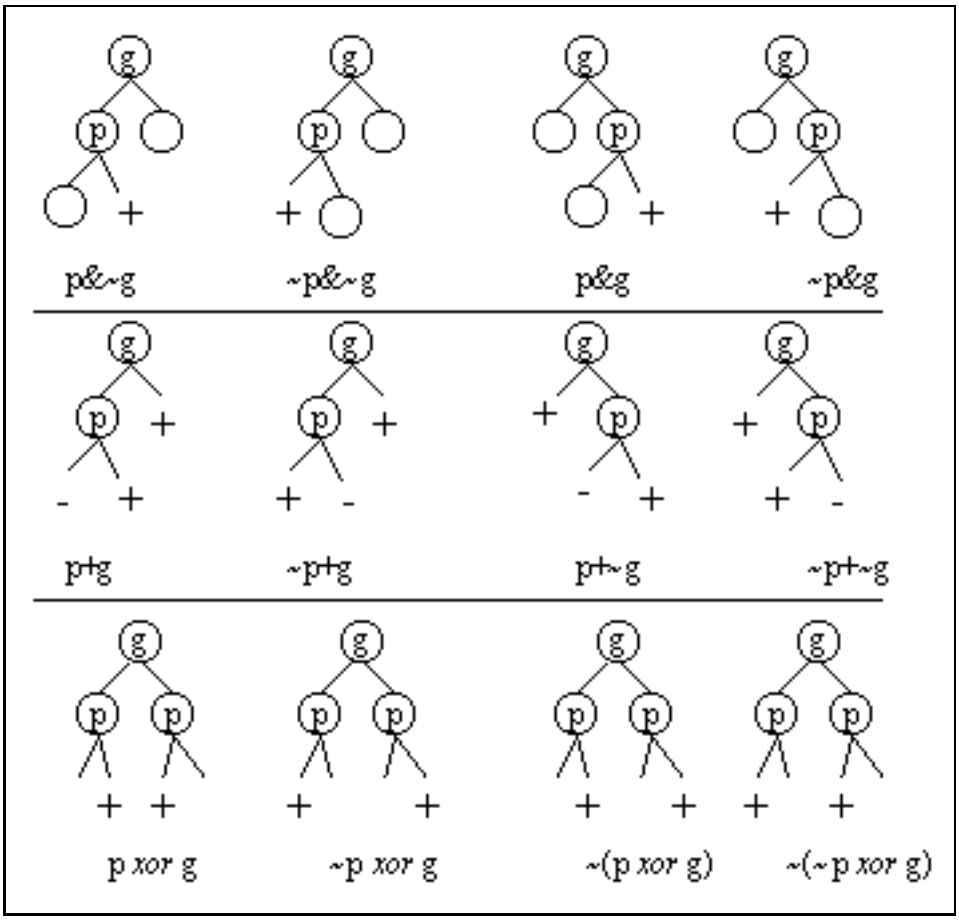

FIG. 8.4 – *Formes détectées à l'aide de l'algorithme FRINGE et ses dérivés*

justement là où l'on veut former les variables synthétiques. Il est à prévoir que les variables synthétiques reflètent ce bruit et que la convergence de l'algorithme soit mal assurée, surtout lorsque le nombre initial d'attributs candidats est très élevé.

– enfin, dernier reproche auquel nous essayerons de répondre dans la section suivante, les formes constituées sont dépendantes du caractère glouton de l'algorithme d'apprentissage. Il est ainsi incapable de détecter les interactions entre variables qui font que deux attributs pris en même temps se révèlent excellents alors que considérés isolément, ils n'engendrent pas des partitions de bonne qualité. L'arbre devient alors exagérément dépendant du fichier d'apprentissage.

En fait, ce type d'exploration se comporte très bien pour une faible catégorie de concepts [Ragavan et Rendell, 1991]. Les différents tests sur des données artificielles, représentant différentes fonctions booléennes, montrent une réduction significative de la taille de l'arbre et une amélioration de la capacité à apprendre sur des petits échantillons. Sur des bases réelles, la réduction persiste, en revanche il n'y a pas d'amélioration significative des performances [Yang *et al.*, 1991b].

8.4 Construction par recherche en avant de combinaisons booléennes de variables: détection de l'interaction

Une des limitations bien connues des graphes d'induction est leur nature gloutonne, une segmentation correspond à un optimum local et est irrévocable. On sait que la recherche de graphe le plus simple qui soit consistant sur les données est NP-complet [Takenaga et Yajima, 1993] [Hyafil et Rivest, 1976] [Murphy et McCraw, 1991], on sait également que la stratégie gloutonne, qui correspond à la projection des observations sur un espace à une dimension, permet de s'en approcher de manière convenable [Murthy et Salzberg, 1995a], on peut se demander si la projection sur un espace de plus grande dimension ne permettrait pas d'améliorer le comportement du classifieur lorsque le concept à décrire est complexe : la stratégie correspondante porte le nom de "recherche en avant"[42] [Sarkar *et al.*, 1994].

Dans cette section, nous discuterons succinctement de la problématique de la recherche en avant, de ses avantages et inconvénients potentiels, nous présenterons ensuite un algorithme qui la traduit sous la forme de construction de variables synthétiques.

42. lookahead search

8.4.1 Recherche en avant : avantages et inconvénients

La meilleure recherche en avant que l'on connaisse consiste à poser toutes les questions possibles pour déterminer l'arbre optimal, [Murphy et Pazzani, 1994] l'ont fait sur une machine massivement parallèle pour évaluer l'effectivité du principe du rasoir d'Occam. Dans la pratique, sur des bases réelles pouvant contenir plusieurs milliers, voire millions, d'observations, surtout avec de nombreux attributs prédictifs candidats, il est clair que cette option n'est guère réaliste.

En revanche, localement, sur un noeud du graphe, de nombreux auteurs pensent que la recherche en avant limitée peut apporter un gain significatif en précision et en concision du graphe [Norton, 1989] [Buntine et Caruana, 1991] [Wallace et Patrick, 1993]. Les arguments ne manquent pas pour justifier cette approche, un des plus intéressants est certainement la prise en compte de l'interaction entre les variables dans la prédiction des classes. Pour illustrer notre propos, nous allons prendre un exemple simple de la fonction XOR. Dans la figure 8.5, nous montrons une topologie qu'un arbre classique avec optimisation locale assortie d'une règle d'arrêt ne saura certainement pas apprendre. On pourrait conjecturer par contre qu'une construction "hurdling", où l'on accepte la segmentation même si elle n'est pas pertinente du point de vue de la mesure d'évaluation dans l'optique de l'élagage, permettrait de s'en affranchir. C'est compter sans la présence des autres attributs qui eux, tout en étant pas du tout informatifs, peuvent entrer au bénéfice du bruit dans la construction de l'arbre. De fait, dans notre exemple (figure 8.5), l'arbre minimum pour restituer le concept peut être perturbé par l'intervention d'un tierce variable X_3 qui, sans perturber la prédiction si l'échantillon est suffisamment grand, entraîne la construction d'un arbre surdimensionné (figure 8.6).

Au revers de la recherche avant, il y a tout d'abord une très forte gourmandise en temps de calcul. Pour une base avec p attributs prédictifs, sur un noeud on effectuera p tests; dans une recherche en avant à un niveau, on aura à tester $p + p(p-1)$ partitions à évaluer; dans une recherche en avant à deux niveaux, ce nombre s'élève à $p + p(p-1) + p(p-1)(p-2)$: le nombre de tests à effectuer est exponentiel au regard du nombre de niveaux de recherche. [Ragavan *et al.*, 1993b] par exemple rapportent qu'il faudrait des mois pour apprendre certaines classes de concepts à l'aide du package IND de [Buntine et Caruana, 1991] sur de puissantes stations de travail.

Plus inquiétante encore est la pathologie associée à la recherche en avant lorsque le niveau d'exploration est mal défini. [Murthy et Salzberg, 1995b] ont rapporté que la recherche en avant à un niveau pouvait dans certains cas produire des arbres nettement moins bons que ceux issus de la recherche gloutonne, ils confirmaient sur bases réelles les résultats obtenus par [Nau, 1983] [Mutchler, 1993] dans des domaines où pourtant les données ne sont pas bruitées (recherche de fin de jeux dans les échecs par exemple). Il est clair dès lors que le choix de la profondeur d'exploration est primordiale, une optimisation sur un espace sous-dimensionné peut "enfer-

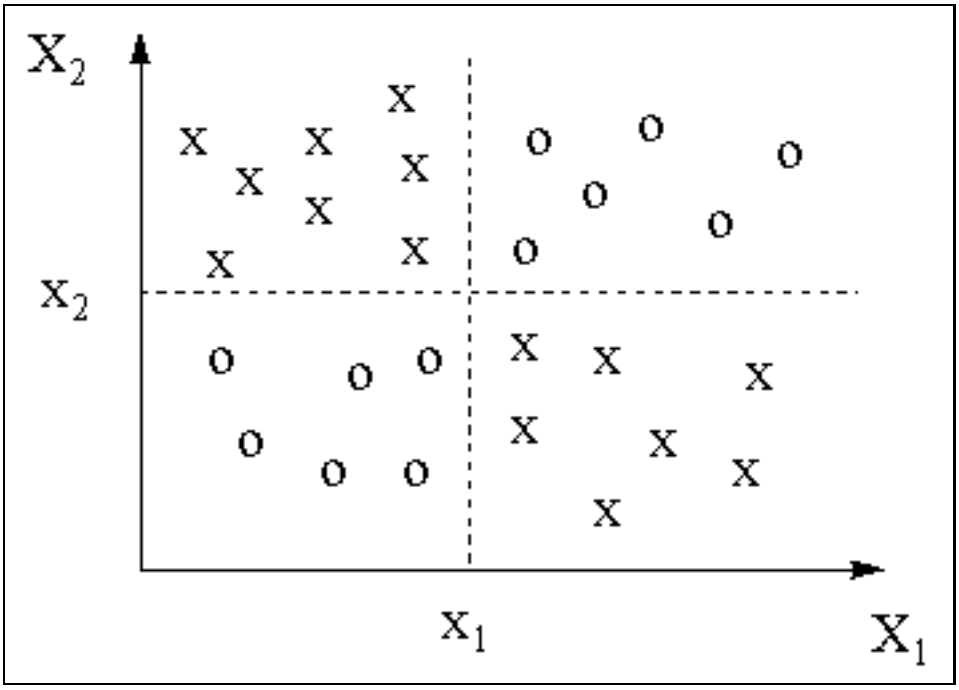

FIG. 8.5 – *Le concept XOR dans un espace à deux dimensions*

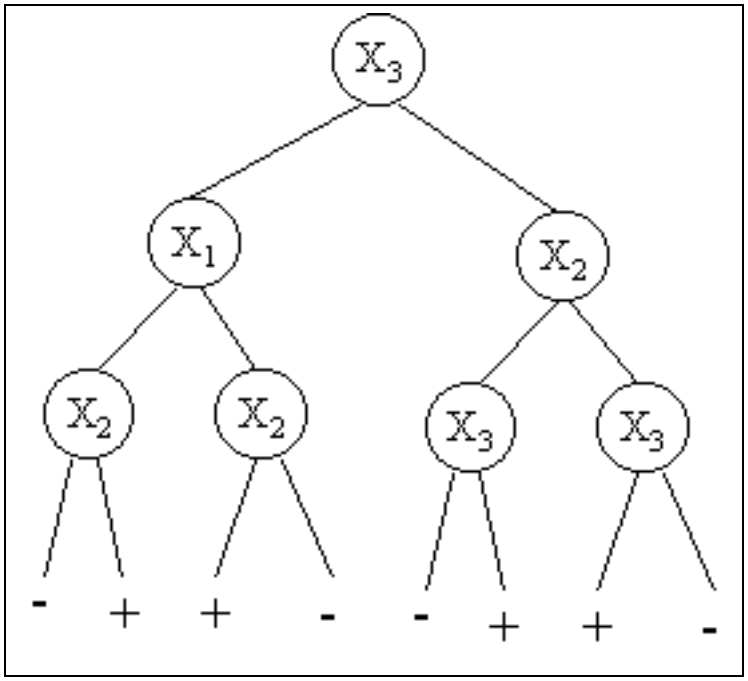

FIG. 8.6 – *Intervention inopportune d'une variable "bruit" X_3 dans un arbre traduisant le concept XOR*

rer" l'algorithme dans des voies de garage qui hypothèquent la suite de l'expansion de l'arbre. [Murthy et Salzberg, 1995b] ont d'ailleurs montré expérimentalement que le couple "glouton-élagage" se comportait mieux en terme de classement que la recherche en avant à un seul niveau.

Au regard de ces différents travaux, le meilleur compromis serait un algorithme qui associerait à la fois la puissance de la recherche sur plusieurs niveaux à une heuristique qui permettrait de n'explorer que les "bonnes" solutions.

8.4.2 L'algorithme L.F.C (Lookahead Feature Construction)

[Ragavan *et al.*, 1993a] affirment que l'association de la construction de variables intermédiaires et de la recherche en avant se révélait plus efficace que l'une de ces techniques isolée, en termes de concision et de précision du classifieur. Ils ont proposé l'algorithme LFC qui construit un arbre de décision en appliquant une recherche en avant limitée contrainte, combinée à la construction de variables synthétiques sur les noeuds.

LFC est un algorithme assez compliqué qui utilise plusieurs "astuces" pour limiter la complexité de la recherche en avant. La base est toujours la stratégie "diviser pour régner" d'ID3 [Quinlan, 1986b], dont il emprunte d'ailleurs la fonction d'évaluation. La principale distinction vient de la sélection de la variable pour la segmentation sur un noeud : on propose ici une recherche en avant que l'on traduit à travers la composition de nouveaux attributs formés de conjonctions de propositions $(X = x)$. La limitation des investigations est réalisée à l'aide des heuristiques suivantes :

- à la différence des méthodes classiques [Norton, 1989] où la recherche est limitée en profondeur, ici on en limite également la largeur. A chaque niveau de recherche, l'algorithme ne retient que les α (fixé par l'utilisateurs) meilleurs attributs (qui peuvent prendre la forme d'une proposition ou d'une conjonction de propositions). L'hypothèse sous-jacente est que les partitions très peu informatives à ce niveau ont très probablement peu de chances d'être améliorées plus tard. Cela permet effectivement de réduire considérablement les temps de calculs, toutefois, nous remarquons que l'on retombe dans les travers de l'exploration gloutonne. Dans notre exemple (figure 8.5), pour peu qu'il y ait d'autres attributs assez bruités dans la base d'apprentissage, les variables X_1 et X_2 ne seront jamais introduits dans les solutions potentielles.

- la deuxième limitation vient de l'observation selon laquelle les sous-groupes de faible entropie circonscrits par une proposition ou une conjonction de propositions ne profitent probablement pas de l'adjonction de nouvelles propositions. Il en résulterait seulement une réduction de l'effectif associé sans amélioration de la pureté.

- la troisième limitation repose sur un filtrage basé sur les couvertures respectives des attri-

buts. Si l'un couvre n_1 individus, on n'acceptera de l'associer avec une proposition que si cette dernière couvre au moins $\lambda \times n_1$ observations (λ est fixé par l'utilisateur, $0 < \lambda < 1$). L'objectif est d'éviter les spécialisations exagérées.

- enfin, la profondeur d'exploration (*lev.*) est fixée par l'utilisateur, dans deux cas on peut pourtant la réduire : lorsque toutes les prémisses décrivent de sous-groupes purs, les étapes suivantes n'ont plus lieu d'être; de même, lorsque l'étape précédente n'a amené aucune amélioration, il est inutile de poursuivre.

8.4.3 Limitations de L.F.C

Appliquée sur des données synthétiques, [Ragavan *et al.*, 1993a] ont montré que leur stratégie surpassait constamment C4.5 [Quinlan, 1993a]. Ces résultats pourtant ont été largement tempérés par une étude exhaustive de [Vilalta *et al.*, 1995] sur l'influence des différentes heuristiques dans le fonctionnement de LFC, on y voit surtout une énorme difficulté à fixer les paramètres idoines :

- le rôle de α est considérable dans le succès de l'algorithme : s'il est trop fort, nous retrouvons la recherche en avant exhaustive, si elle est trop faible ($\alpha = 1$), nous retombons dans l'algorithme glouton. Dans la pratique, trouver le juste milieu passe par la détection préalable des attributs non-pertinents.

- les performances de l'algorithme reposent en grande partie sur le choix de la profondeur d'exploration (*lev.*), qui elle-même dépend de la complexité du concept à étudier. Pour la fonction XOR par exemple, la valeur 2 suffit, mais il est évident que sur des données réelles, on ne peut le prévoir de manière certaine.

Avec les expérimentations de [Yang *et al.*, 1991a] et de [Vilalta *et al.*, 1995] sur des bases réelles, nous retrouvons les remarques avancées dans la section précédente : à savoir que l'on observe bien une réduction des arbres, en revanche il n'y a pas d'améliorations significatives en performances, sur certains cas on observe même une dégradation des taux de succès. Ces derniers l'attribuent avant tout à la restriction de l'algorithme à des problèmes à deux classes et à des attributs booléens, l'adaptation sur des données réelles se fait au détriment de son efficacité. Pour notre part, nous pensons également que la sensibilité de l'algorithme à ses paramètres n'est pas étranger à ces difficultés.

8.5 Cas particulier de l'espace de représentation continu : la préparation statistique des données

Les domaines où tout ou partie des attributs prédictifs sont de nature continue constituent un champ privilégié de la "feature construction ". En effet, les graphes d'induction en apprentissage automatique ont été élaborés à l'origine pour des attributs qualitatifs [Quinlan, 1979][Tounissoux, 1980], et méconnaissent la spécificité de l'espace de représentation numérique :

- les mesures de qualité de partitions sont fondées sur les distributions inconditionnelles et conditionnelles des classes, la distribution des exemples dans l'espace de représentation est passée sous silence;
- un espace de représentation euclidien permet de calculer moultes mesures de distances qui rendent compte de la proximité entre individus;
- il existe de nombreuses méthodes de structuration qui permettent de faire émerger des catégories "naturelles" plus ou moins homogènes du point de vue de la variable à prédire.

[de Merckt, 1993] est un des rares chercheurs à avoir présenté une alternative à la construction de variables intermédiaires dans un espace numérique. Son idée était de transformer la mesure de qualité de segmentation en introduisant la notion de *contraste* qui correspondait tout simplement au rapport entre l'inertie inter-groupes et l'entropie conditionnelle. Dans la pratique, cette voie s'avère peu décisive pour plusieurs raisons :

- elle échoue nettement lorsque la distribution des classes est multimodale [Murthy, 1995].
- elle ne permet pas de dépasser une des limitations inhérentes aux graphes : sa nature gloutonne. Même si l'on tient compte de la proximité dans la phase de discrétisation, les variables sont considérées unes à unes, toute interaction est ignorée. Avec les méthodes polythétiques de la classification automatique, ou encore les techniques descriptives de l'analyse de données, de telles limitations sont facilement levées.

A la différence des techniques décrites précédemment, où les variables intermédiaires étaient formées à partir d'hypothèses produites par l'algorithme d'apprentissage [43], les nouveaux attributs ici sont construits dans un processus à part que l'on peut considérer comme une phase de pré-traitement des données[44] [Wnek et Michalski, 1994]. Dans ce qui suit, nous présentons deux problématiques de la construction de variables synthétiques pour les graphes d'induction dans un espace continu, nous proposerons alors une approche simple et peu coûteuse que nous évaluerons à travers une large expérimentation sur nos bases benchmarks.

43. Hypothesis-driven constructive induction
44. Data-driven constructive induction

8.5.1 Problématiques de la construction de variables synthétiques dans le cadre des graphes d'induction

Nous voulons construire une ou plusieurs variables synthétiques qui traduisent la "structure" du nuage de points plongé dans IR^p (proximité, groupes naturels...), nous voulons savoir si elles sont utiles pour la prédiction de la classe. Les principales méthodes connues aujourd'hui exploitent plus ou moins la notion de séparabilité.

Construction par classification automatique

La classification automatique vise à créer, à partir d'un ensemble d'observations, plusieurs groupes tels que les membres de chaque groupe diffèrent d'un autre aussi peu que possible au regard d'un critère donné. L'idée sous-jacente dans la constitution de variables synthétiques est de faire émerger une nouvelle variable qui exploite les informations de proximité dans l'espace de représentation [Oliver et Dowe, 1995]. Le principe est simple : on procède à une classification automatique, le nouvel attribut est alors une variable qualitative qui contient autant de modalités qu'il y a de groupes dans l'analyse précédente. Lors de la phase de généralisation, on affecte l'individu à classer au groupe dont il est le plus proche (au sens du critère de proximité couramment utilisé en analyse typologique : le plus proche voisin, le voisin le plus éloigé, la distance au centre de gravité ... [Chandon et Pinson, 1981]) avant d'exploiter le classifieur extrait de l'apprentissage.

Dans cette sous-section, nous allons nous intéresser plus particulièrement aux travaux de [Murthy, 1995] qui a procédé à des expérimentations à grande échelle pour éprouver sa méthode et dont la philosophie nous permet de comprendre la problématique sous-jacente de la construction de variables intermédiaires dans un espace continu.

Il existe une multitude de méthodes de classification automatique, [Murthy, 1995] préfère la stratégie fondée sur l'*arbre de longueur minimale* parce qu'elle est rapide et a démontré au fil des différentes expérimentations son efficacité et sa fiabilité [Graham et Hell, Annals of History of Computing] [Zahn, 1971]. Le principe est simple : il s'agit de construire un graphe connexe $G(V, E)$, non-orienté, qui connecte toutes les observations dans l'espace de représentation et dont la longueur est minimale. V représente l'ensemble des sommets (individus), et E les arêtes qui les relient. "Longueur" doit être comprise dans le sens "somme des longueurs de toutes les arêtes". Il peut y avoir un grand nombre de solutions, l'algorithme de Kruskall (1954) assure la découverte de l'une d'entre elles.

De manière générale, en classification automatique, les groupes sont composés en coupant les arêtes les plus longues parmi E. Dans la cadre particulier de la préparation des données pour l'induction, [Murthy et Salzberg, 1992] proposent d'éliminer les arêtes qui relient des sommets d'étiquettes différentes. Ainsi, nous sommes assurés d'avoir des groupes homogènes. Dans la figure

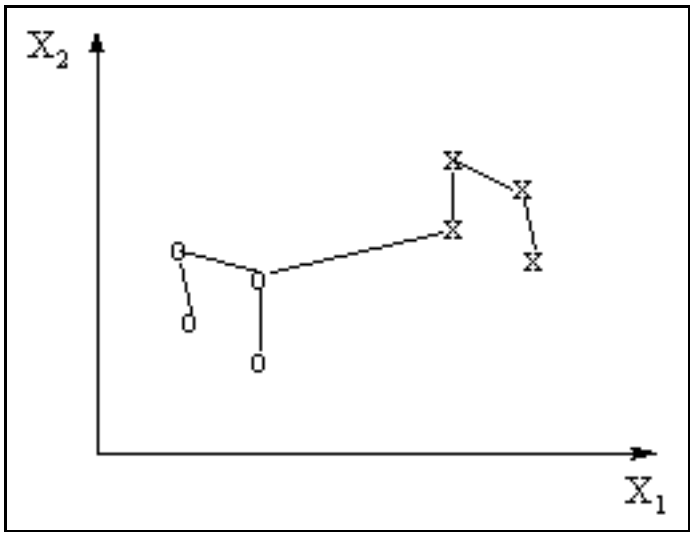

FIG. 8.7 – *Arbre de longueur minimale*

8.7, nous avons représenté l'arbre de longueur minimale pour un exemple à deux classes $\{x, o\}$ dans un espace à deux dimensions, l'arête la plus longue ici relie deux individus d'étiquettes différentes, la connexion est supprimée.

En introduisant parmi les attributs prédictifs, une nouvelle variable issue de cette phase de pré-traitement, [Murthy, 1995] montre à partir d'expérimentations sur données réelles et synthétiques que l'on peut réduire considérablement la taille du graphe et dans certains cas améliorer les performances du classifieur. Il rejoint ainsi les résultats mis en avant dans [Oliver et Dowe, 1995] où la classification est élaborée à l'aide d'un algorithme différent (SNOB [Wallace et Dowe, 1994]).

Mais revenons à la spécification de la classification proposée par [Murthy et Salzberg, 1992]. Nous supprimons les connections entre deux sommets d'étiquettes différentes. Que signifie cette suppression? Si toutes les connections sont coupées, cela veut dire que les étiquettes sont complètement mélangées dans l'espace de représentation, et qu'il n'y a pas de structuration naturelle des données. De fait, chaque groupe qui formera une modalité de la nouvelle variable ne comporte qu'un seul individu. Il est à prévoir que la nouvelle variable n'apporte guère d'information pour la prédiction de la variable endogène. Dans le cas contraire, où le nombre d'arêtes coupées est égal au nombre de classes du problème moins un, chaque groupe est naturellement pur et contient tous les représentants d'une classe. De fait, il y a équivalence entre les groupes de la classification et les classes de la variable à prédire $Y(.)$, un arbre à un seul niveau suffit pour discriminer complètement la population.

[Sebban, 1996] a travaillé sur la même spécification pour définir le problème de la "séparabilité". En calculant le nombre théorique d'arêtes supprimées dans une hypothèse de distribution aléatoire des observations, il a réussi à caractériser les situations dans lesquelles on peut conclure à l'existence de structures dans la disposition des individus dans l'espace de représentation. C'est

le cas lorsque le nombre d'arêtes supprimées est significativement faible.

Le problème dual du comptage des arêtes enlevées (u) est tout simplement le dénombrement du nombre de groupes constitués (r) que l'on nomme "amas", avec la relation $r = u + 1$. On considère que la variable synthétique est pertinente si le nombre d'amas est suffisamment faible. Si nous travaillons dans IR nous retombons sur la notion de "séquences".

Construction par projection

Nous voulons cette fois-ci construire une ou plusieurs variables synthétiques qui soient la résultante de transformations algébriques sur les attributs continus initiaux. En effet, aussi séduisante que soit la constitution de la variable par classification automatique, elle impose des contraintes calculatoires qui semblent rédhibitoires dans la pratique. Pour un nouvel individu à classer, nous sommes obligés de calculer sa distance vis-à-vis de tous les points de l'échantillon d'apprentissage pour déterminer son groupe : si la taille de ce dernier est très élevé, et qu'il y a beaucoup d'individus à classer, il est clair que le processus s'avérera très gourmand en ressources mémoire (sauvegarde de toutes les coordonnées) et en temps de calcul.

Le cas idéal de la transformation algébrique est représenté par l'exemple de la location d'appartements où on dispose de leurs "longueur" et "largeur" pour prédire les classes de prix, il suffit de construire le nouvel attribut "surface" où

$$surface = longueur \times largeur$$

pour construire un classifieur satisfaisant[45]. En classement, il suffit d'appliquer la formule pour disposer de la coordonnée de l'individu dans le nouvel espace de représentation.

Comment juger la qualité de la nouvelle variable? La meilleure piste nous est donnée dans la sous-section précédente : nous voulons une variable telle que les classes soient au mieux discriminées, ou encore séparables, le long de son axe de représentation. Dans IR^p, nous avions caractérisé ce sujet à l'aide du concept "amas", dans IR il paraît logique de passer à la notion de séquences que nous développons en longueur dans le chapitre sur la discrétisation. Ce rapprochement n'est pas innocent car le nouvel attribut est continu et sera discrétisé en deux ou plusieurs intervalles dans la phase d'apprentissage. De fait, nous pouvons poser la problématique suivante de la construction de variables synthétiques : "*on recherche une nouvelle variable telle que la projection des individus sur cet axe engendre un nombre minimum de séquences*".

La première difficulté est dans la détermination de la forme de la fonction de transformation. Il est évident que l'on ne peut pas tester toutes les possibilités, nous devons donc nous restreindre à une catégorie de fonctions. Sans présumer de ses qualités, il est clair que l'analyse linéaire reste un préalable de choix pour une première exploration, d'autant que l'analyse de données regorge de

45. avec une certaine erreur quand même puisque tous les appartements ne sont pas rectangulaires

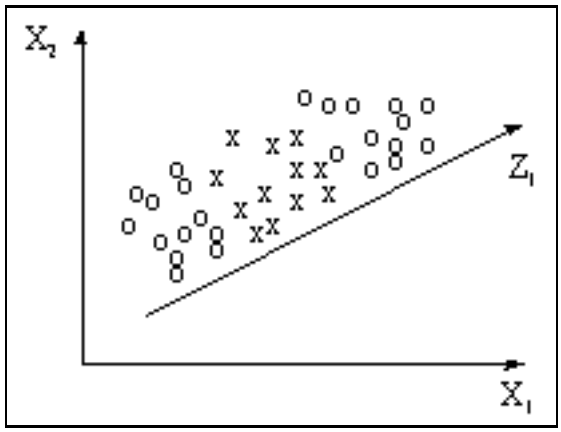

FIG. 8.8 – *La projection sur l'axe Z_1 permet une discrétisation parfaite en trois intervalles*

techniques de description linéaires. Néanmoins, nous devons avouer que, même en se restreignant à ce type de fonctions, nous n'avons pu trouver de solution déterministe à la problématique ci-dessus. Dans la sous-section suivante, nous présentons une heuristique qui, bien que très peu satisfaisante par rapport à nos préoccupations, donne des résultats intéressants.

8.5.2 Expérimentations sur une approche naïve de la construction de variables synthétiques

Les axes factoriels de l'analyse en composantes principales

La première solution que nous avons essayée a été l'analyse discriminante descriptive. Cette dernière vise à construire une ou plusieurs variables (les axes factoriels) tels que l'inertie inter-classes des observations soit maximale. C'est une procédure supervisée, elle représente en quelque sorte la généralisation multidimensionnelle de la solution préconisée par [de Merckt, 1993]. Hélas dans les faits [Yip et Webb, 1994], cette procédure se révèle peu efficace parce qu'elle n'arrive pas à traiter convenablement les cas de distributions multimodales des classes. Dans l'exemple de la figure 8.8, l'axe Z_1 indiscernable par l'analyse discriminante permet pourtant de mettre en évidence 3 séquences correspondant à la solution optimale.

Pour notre part, nous avons voulu tester le comportement de variables en provenance de l'analyse en composantes principales. Ce choix a été motivé par différentes raisons qui peuvent paraître de prime abord contradictoires ou sans rapport avec notre problématique originelle :

- l'analyse en composantes principales est non supervisée i.e ne tient pas compte des classes, de fait elle n'est pas sensible aux distributions multimodales (dans le cas de la figure 8.8, elle aurait trouvé l'axe Z_1).

- elle propose de nombreuses aides à l'interprétation dont on pourra tirer profit par la suite

pour réduire les expressions des axes factoriels (e.g si deux variables sont fortement corrélées avec un axe, il suffit de n'en sélectionner qu'une dans l'équation de la nouvelle variable).

– elle propose plusieurs solutions que l'on sait interpréter :sur le premier axe, la dispersion des individus est maximum; sur le dernier, elle est minimum.

Ne nous voilons pas la face, il faudrait un grand hasard pour que l'un des axes factoriels corresponde à la solution recherchée. En revanche, il est clair qu'ils traduisent la topologie des observations dans l'espace de représentation continu, et en ce sens constituent une source d'informations non-négligeable. Du reste, il a été appliqué avec succès dans des problématiques très spécifiques telles que la reconnaissance des caractères manuscrits [Grother, 1992].

Expérimentation

Dans cette expérimentation, nous n'utiliserons que les bases contenant plusieurs attributs prédictifs continus. Afin d'évaluer le rôle du pré-traitement par l'analyse en composantes principales, nous adopterons le protocole suivant :

1. nous effectuons 10 répétitions du couple apprentissage validation avec des proportions de 60% et 40%;

2. nous effectuons de nouveau 10 répétitions avec les mêmes caractéristiques mais pour lesquelles, dans la portion d'apprentissage, nous créons plusieurs variables à l'aide de l'ACP qui seront candidates à la construction du classifieur. Dans la phase de validation, les individus sont projetés comme observations supplémentaires dans le nouvel espace de représentation, avant d'être étiquettés à l'aide du classifieur construit en apprentissage.

La production des variables est faite suivant la procédure décrite ci-après. Soit $\aleph = (X_a|\ldots|X_h)$ la matrice formée par les attributs prédictifs continus, nous calculons la matrice centrée réduite $\aleph_{cr}$. Nous formons la matrice de variance covariance $V = \frac{1}{n}\aleph'_{cr}\aleph_{cr}$ à partir de laquelle nous extrayons la matrice des vecteurs propres U. Les nouvelles variables candidates sont alors les projections sur les axes factoriels $Y = (Y_a|...|Y_h)$ tel que $Y = \aleph_{cr}U$.

Les moyennes des taux de succès et des tailles de bases de règles afférentes sont recensées dans les tables 8.4 et 8.5. Les écarts significatifs pour un test de Student pour échantillons appariés à 5% sont signalés. Nous avons utilisé la méthode ID3, avec un test du χ^2 comme règle d'arrêt d'expansion du graphe.

Les résultats amènent plusieurs réflexions :

– en terme de précision, l'introduction d'une série de variables peu ou prou pertinentes n'amène pas beaucoup d'amélioration. On remarque que les deux seuls fichiers pour les-

Base	Sans Nouvelles Var.(1)	Avec Nouvelles Var.(2)	Signif
autos	0.61 ± 0.048	0.55 ± 0.059	$(1) > (2)*$
breast-c	0.95 ± 0.015	0.96 ± 0.008	$(1) < (2)*$
cpuperf	0.93 ± 0.039	0.93 ± 0.043	
credit	0.90 ± 0.013	0.90 ± 0.011	
flags	0.53 ± 0.097	0.50 ± 0.059	
hepatitis	0.79 ± 0.046	0.78 ± 0.060	
ionosphere	0.87 ± 0.029	0.87 ± 0.025	
iris	0.95 ± 0.020	0.95 ± 0.025	
pima	0.74 ± 0.009	0.73 ± 0.023	
wave	0.70 ± 0.055	0.76 ± 0.025	$(1) < (2) **$
wine	0.92 ± 0.030	0.90 ± 0.039	$(1) > (2)*$

TAB. 8.4 – *Taux de succès en validation - Sans et avec nouvelles variables contruites par ACP*

Base	Sans Nouvelles Var.(1)	Avec Nouvelles Var.(2)	Signif
autos	11.4 ± 1.71	12.5 ± 2.01	
breast-c	13.4 ± 1.50	11.3 ± 1.25	$(1) > (2) **$
cpuperf	5.5 ± 0.53	6.5 ± 0.71	$(1) < (2) **$
credit	4.6 ± 2.11	4.8 ± 2.20	
flags	10.4 ± 3.72	10.1 ± 3.57	
hepatitis	4.6 ± 1.17	4.5 ± 0.85	
ionosphere	12.6 ± 1.26	10.5 ± 1.58	$(1) > (2) ***$
iris	4.6 ± 0.84	4.9 ± 0.87	
pima	18.4 ± 5.03	23.2 ± 6.61	$(1) < (2) ***$
wave	16.3 ± 1.94	13.8 ± 1.68	$(1) > (2) **$
wine	6.0 ± 0.94	5.2 ± 0.63	$(1) > (2)*$

TAB. 8.5 – *Nombre de règles - Sans et avec nouvelles variables construites par ACP*

quels l'avancée est significative sont entièrement composés d'attributs prédictifs continus (Breastc, Wave).

- ce bilan reste d'actualité en ce qui concerne la taille des bases de règles construite. La réduction est conséquente pour les bases à très forte majorité d'attributs continus (Breatsc, Ionosphere, Wave, Wine), elle est d'autant plus forte lorsque le concept est complexe (ce que l'on peut détecter avec le nombre moyen de règles : Ionosphere et Wave).

- plus étonnant sont en revanche les dégradations que l'on observe sur certains fichiers, tant en précision (Autos, Wine) qu'en complexité (Cpuperf, Pima !). Elles montrent une fois de plus que l'on ne peut introduire impunément n'importe quelle variable dans l'élaboration des graphes d'induction, ces derniers sont sensibles aux attributs non-pertinents malgré que l'on ait choisi sciemment une méthode qui n'adopte pas le principe de la construction "hurdling" [Almuallim et Dietterich, 1992][Imam, 1995].

Au final, le bilan de création de nouvelles variables par l'ACP est assez mitigé. D'une part, il est clair que dans certains cas, l'apport de nouvelles variables est un plus non négligeable; d'autre part, il faut faire attention à ne pas amener des attributs non pertinents car ils risquent de détériorer la qualité du classifieur. Mais comment peut-on en prévoir à l'avance la qualité? Nous pensons, et nos travaux vont activement en ce sens, que la problématique que nous avons proposée ci-dessus (cf. 8.5.1.0) est une piste très intéressante, encore faut-il arriver à trouver soit une solution directe soit des heuristiques qui nous permettent de s'en approcher.

De toute manière, il est clair que cette voie de recherche reste encore à défricher, ne serait-ce que pour la multitude de fonctions que l'on peut tester. Le linéaire n'est qu'une approche préliminaire, l'exploration de fonctions plus riches de manière semi-automatique, soit à l'aide d'experts, soit à l'aide de procédés visuels représentent des compromis assez prometteurs.

8.6 Conclusion

Les procédés décrits dans ce chapitre, à la différence des graphes, visent à pallier les faiblesses des arbres en matière de pouvoir de représentation en créant des variables intermédiaires plus riches. Ici également, les avantages sont manifestes sur données simulées, sur les bases réelles benchmark, on constate généralement une réduction normale de la taille de l'arbre mais pas une amélioration des performances.

Des auteurs comme [Yang *et al.*, 1991a] affirment que ces avantages sont effectifs lorsque la taille de l'échantillon est relativement faible : pas trop car dans ce cas le concept est insuffisamment décrit par les observations; lorsque les observations sont au contraire abondantes, la fragmentation n'est plus un handicap. C'est un argument discutable parce que d'une part il est

extrêmement difficile de déterminer quelle est cette taille, et d'autre part, nous avons rarement le choix de la taille de l'échantillon dans la réalité, sauf cas particulier.

Les méthodes décrites dans ce chapitre ne couvrent pas toutes les options de transformations des données, il en existe un grand nombre mais toutes répondent au même objectif, trouver des variables intermédiaires qui saisissent des concepts que l'on décrit difficilement avec les arbres [Murphy et Pazzani, 1991].

Chapitre 9

Discrétisation des attributs continus

9.1 Introduction

La plupart des méthodes symboliques d'induction de règles à partir d'exemples [Quinlan, 1979] [Michalski et Larson, 1983] [Clark et Niblett, 1989] ont été conçues pour des variables de type catégoriel prenant leurs valeurs dans un ensemble de cardinal fini. Par exemple, le sexe ne peut être que masculin ou féminin. Lors de l'introduction de variables de type continu, prenant leurs valeurs dans l'ensemble des réels, de toute manière possédant une structure d'ordre, il est nécessaire de les transformer afin de les rendre compatibles avec les algorithmes d'apprentissage. Le processus de découpage d'un attribut de type continu en un ensemble d'intervalles disjoints est désigné sous le terme de discrétisation.

Les chercheurs ont réalisé l'importance de ce domaine de recherche assez récemment. L'extension et le développement de calculateurs rapides ont de plus en plus amené les algorithmes d'apprentissage à appréhender des applications où les données étaient de n'importe quel type, sans préparation préalable. Les premières méthodes de découpage de données étaient relativement simples, peu d'études ont été initiées pour évaluer leur effet sur l'algorithme d'apprentissage [Weiss et Kulikowski, 1991]. Dans le cadre particulier des graphes d'induction, le découpage local binaire des variables maximisant la mesure de qualité de partition semblait naturel au point que les auteurs n'en fasse pas explicitement mention dans leur travaux [Bouroche et Tenenhaus, 1970] [Breiman *et al.*, 1984]. Pourtant, cette étape de pré-traitement des données est importante. En effet, elle fait partie de la phase d'apprentissage, on essaie d'estimer des bornes de discrimination entre les individus à partir d'un échantillon d'apprentissage. De fait, elle conditionne le choix des attributs discriminants lors de la construction du modèle prédictif : un mauvais choix des points de découpage peuvent faire perdre de l'information [Celeux et Robert, 1993] au point d'hypothéquer complètement les performances du classifieur [de Merckt et Quinlan, 1996].

Depuis le début des années 90, la discrétisation est devenue un sujet d'étude très prisé. Les

principaux travaux de synthèse sont l'oeuvre de [Rabaseda *et al.*, 1996b] et [Dougherty *et al.*, 1995]. De manière générale, on distingue quatre thèmes de débat :

- *méthode supervisée contre méthode non-supervisée* : dans la discrétisation supervisée (contextuelle), on tient compte explicitement de la distribution des classes lors du découpage de l'attribut prédictif. En non supervisé en revanche (non-contextuelle), on ne tient compte que de la similarité des individus dans $\mathbb{R}$ sans se préoccuper de leur étiquette.

- *discrétisation locale contre discrétisation globale* : la discrétisation globale, que l'on pourrait encore qualifier d'a priori [Richeldi et Rossotto, 1995], consiste à transformer la variable continue dans une phase de pré-traitement des données, les bornes définies sont ainsi fixées à l'avance, elles ne seront plus remises en cause dans la mise en oeuvre de l'algorithme d'apprentissage. Dans le cadre particulier des graphes d'induction, l'objectif étant de produire à chaque noeud une partition disjointe des individus, il est possible de procéder à une discrétisation locale "à la volée" [Wehenkel, 1997], simplement binaire ou à plusieurs intervalles.

- *bornes de discrétisation "dures" contre bornes de discrétisation "molles"*: d'une manière générale, les méthodes de discrétisation cherchent à déterminer des bornes qui sont fixées une fois pour toutes et utilisées telles quelles en généralisation. Or, cette borne a été estimée sur un fichier d'apprentissage, la valeur calculée est certainement entâchée d'une certaine imprécision, il serait plus appropriée de dire : "la borne est autour de la valeur d", tout le problème étant par la suite de quantifier la latitude que l'on donne au terme "autour". [Carter et Catlett, 1987] ont certainement été parmi les premiers à discuter de l'opportunité du passage à des bornes "molles" qui conduisaient ainsi à des arbres dans lesquels les individus pouvaient emprunter plusieurs chemins, les feuilles contenant ainsi des individus fractionnaires. Par la suite, l'adoption d'un cadre théorique rigoureux a permis une étude plus poussée fondée sur la théorie des ensembles flous [Marsala, 1996]. Dans ce chapitre, nous n'aborderons pas la construction des arbres flous, nous noterons cependant que c'est une voie de recherche qui semble assez intéressante ne serait-ce que pour réduire la variance de l'estimation de la borne de discrétisation [Wehenkel, 1997].

- *discrétisation statique contre discrétisation dynamique* : les méthodes statiques recherchent les bornes de découpages des variables les unes indépendamment des autres, c'est la stratégie la plus couramment utilisée dans la construction des graphes d'induction. Les méthodes dynamiques essaient de découper toutes les variables prédictives simultanément, par exemple en procédant à une classification dans $\mathbb{R}^p$ afin de définir des prototypes que l'on substituera aux attributs continus dans l'induction [Rauber *et al.*, 1994]. Même si l'on

peut considérer cette voie comme très prometteuse [Kohavi et Sahami, 1996], nous ne l'explorerons pas dans ce chapitre car elle s'éloigne de la philosophie des graphes d'induction qui visent à élaborer des classifieurs compréhensibles à l'homme.

Les thèmes précédents nous ont largement suggéré l'organisation de ce chapitre. Dans un premier temps, nous présenterons une formalisation précise du problème de la discrétisation. Puis, nous discuterons de l'opportunité de l'usage d'un test de séparabilité comme préalable à la discrétisation. La section suivante sera consacrée au débat supervisé-non-supervisé, nous y étudierons plus particulièrement l'influence des mesures de qualité des partitions. Nous aborderons alors la discrétisation en L ($L \geq 2$) intervalles. Nous accordons une importance particulière à cette stratégie car, à la différence de la discrétisation binaire, elle peut s'appliquer à n'importe quel algorithme d'induction de règles. Dans le cadre de notre travail, nous avons d'ailleurs implémenté plusieurs algorithmes de discrétisation, dont un optimal au sens du critère utilisé. La section qui suit sera vouée à l'étude des avantages et inconvénients respectifs des stratégies locales et globales. Puis nous aborderons l'aspect statistique de l'évaluation des points de découpage. Enfin nous conclurons ce chapitre en évaluant les perspectives d'évolution de la recherche dans le domaine de la discrétisation.

9.2 Position du problème et définitions

9.2.1 Formalisation de la discrétisation

Soit D_X le domaine de définition de l'attribut continu $X(.)$. Discrétiser la variable $X(.)$ revient à découper D_X en L intervalles I_l $(l = 1, \ldots, L)$; $(L \geq 1)$ qui seront numérotés de $1, \ldots, L$.

$$\begin{aligned} I_1 &= [d_0, d_1[\\ &\vdots \\ I_l &= [d_{l-1}, d_l[\\ &\vdots \\ I_L &= [d_{L-1}, d_L[\end{aligned}$$

Cela consiste à déterminer les points de discrétisation d_l avec $l = 1, \ldots, L-1$. Une fois ces valeurs trouvées, la variable quantitative $X(.)$ est remplacée par une variable $\widetilde{X}(.)$ qualitative qui prendra ses valeurs dans l'ensemble $\{1, \ldots, L\}$. Ainsi, pour tout individu ω issu de la population Ω,

$$d_{l-1} \leq X(\omega) < d_l \Rightarrow \widetilde{X}(\omega) = l$$

Dans le cas de l'apprentissage supervisé qui nous intéresse ici, les méthodes proposées cherchent les points de discrétisation de la variable $X(.)$ en tenant compte des valeurs prises par un

attribut particulier $Y(.)$, défini sur $\{y_1, ..., y_K\}$. Dans la mesure où le but final est de construire un modèle qui permet de calculer les valeurs de $Y(.)$ en fonction de $X(.)$ il semble naturel de rechercher des points de discrétisation qui nous rapprochent de cette situation. Ainsi, nous pouvons formaliser de la manière suivante l'objectif de la discrétisation :

- *Il s'agit de découper le domaine de définition D_X de $X(.)$, en L intervalles I_l ($l = 1, \ldots, L$ avec $L \geq 1$).*

$$\begin{aligned} I_1 &= [d_{min}, d_1[\\ &\vdots \\ I_l &= [d_{l-1}, d_l[\\ &\vdots \\ I_L &= [d_{L-1}, d_{max}[\end{aligned}$$

Tels que

$$\forall I_l (l = 1, \ldots, L), \exists y_k \in \{y_1, \ldots, y_K\}; P(y_k/I_l) \approx 1$$

Autrement dit, à l'image de la problématique de construction des graphes eux-même, nous cherchons une partition des individus qui minimise le taux d'erreur en classement dans la population. Dans le cas idéal, chaque intervalle de la discrétisation devra contenir exclusivement des individus qui appartiennent à une même classe. Nous noterons que si $L = 1$, cela signifie que l'algorithme de discrétisation n'aura pas pu mettre en évidence un découpage intéressant.

9.2.2 Quelques définitions

Points frontières

Soit $X(\Omega) = \{x_1, \ldots, x_l, x_{l+1}, \ldots, x_a\}$ l'ensemble des valeurs ordonnées de $X(.)$ observées sur la population Ω.

$$x_1 < \ldots < x_l < x_{l+1} < \ldots < x_a$$

On note Ω_l et Ω_{j+1} l'ensemble des individus de Ω ayant pris respectivement la valeur x_l et x_{j+1} sur la variable $X(.)$:

$$\begin{aligned} \Omega_l &= \{\omega \in \Omega \quad / \quad X(\omega) = x_l\} \\ \Omega_{l+1} &= \{\omega \in \Omega \quad / \quad X(\omega) = x_{l+1}\} \end{aligned}$$

On définit d_l le point situé entre x_l et x_{j+1}

$$d_l = \rho * x_l + (1 - \rho) * x_{l+1} \;\; ; \;\; 0 < \rho < 1$$

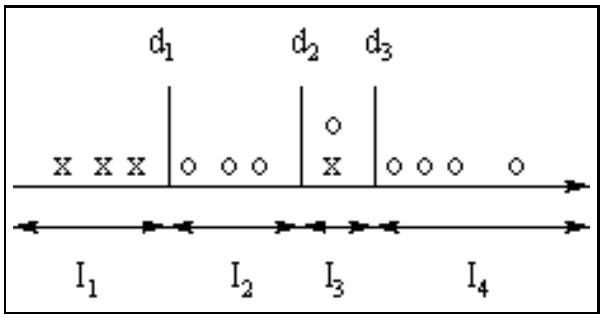

FIG. 9.1 – *Pour un échantillon qui comporte 4 points de la classe "x" et 8 de "o", les points frontières d_1, d_2 et d_3 induisent les intervalles I_1 à I_4, qui constituent une partition de Ω*

Généralement, nous fixons $\rho = 0.5$. On dira que d_l est un point frontière si les classes des individus appartenant Ω_l ne sont pas toutes les mêmes que les classes des individus de Ω_{j+1}, c'est-à-dire :

$$\exists\ \omega \in \Omega_l\ \exists\ \omega' \in \Omega_{l+1}\ telque\ Y(\omega) \neq Y(\omega')$$

On notera U l'ensemble des points frontières ainsi définis, $u = Card(U)$, il induit $u+1$ intervalles que nous noterons $(I_1, I_2, ..., I_{u+1})$.

Séquences

On notera R_l l'ensemble des individus qui se trouvent dans l'intervalle I_l défini par les points frontières, $R_l = \{\omega \in \Omega\ /\ X(\omega) \in I_l\}$. On appelle séquences l'ensemble $\{R_1, ..., R_{u+1}\}$ qui forme une partition de Ω et on pose $r = (u+1)$. Nous représentons dans le graphique 9.1, une disposition possible pour un échantillon de 12 individus, et deux classes (**x** et **o**).

9.3 Test de séparabilité des individus dans IR

Notre objectif est de trouver un ou plusieurs points de discrétisation qui permettent de distinguer au mieux les différentes occurrences de la classe. Avant de procéder à cette partition, il semble pertinent de poser la question suivante : "l'observation des valeurs respectives de l'attribut $X(.)$ pour chaque modalité de l'endogène $Y(.)$ est-elle homogène sur la population parente?". En termes statistiques, nous sommes donc amenés à construire un test statistique dont l'hypothèse nulle s'écrit :

$$H_0 : F(X/Y = y_1) = \cdots = F(X/Y = y_K)$$

9.3.1 Inadéquation des tests "classiques"

Cette formulation n'est pas sans rappeler la problématique de l'analyse de variance, il existe plusieurs tests paramétriques qui la résolvent de manière satisfaisante [Guenther, 1966]. Hélas

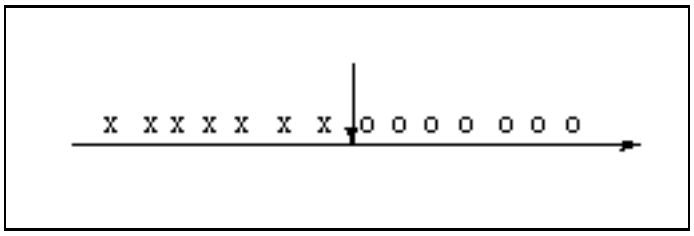

FIG. 9.2 – *Les "x" sont parfaitement séparables des "o"*

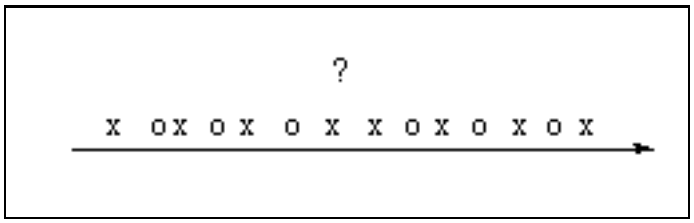

FIG. 9.3 – *Les "x" et les "o" sont complètement mélangés*

ces tests reposent sur une hypothétique distribution normale des observations, peu crédible dans la pratique. Force est de nous diriger vers les tests non-paramétriques, moins puissants certes, mais autrement plus robustes.

Parmi les tests non-paramétriques [Bulle, 1990], sans entrer dans les détails, les tests fondés sur les rangs moyens (White, Kruskall et Wallis ...) souffrent de la nécessité d'hypothèses supplémentaires peu soutenables dans le cadre précis de la discrétisation comme la symétrie des distributions et l'égalité des échelles de mesures. Nous aimerions que notre test soit opérationnel dans les différents cas de figures suivants :

- aucune hypothèse sur la loi de distribution des observations;
- les distributions peuvent être non symétriques et multimodales;
- chaque fonction de répartition peut avoir un facteur d'échelle différent (par exemple, il est possible que la taille soit plus dispersée chez l'homme que chez la femme).

9.3.2 Tests fondés sur les séquences

La série de graphiques suivants nous permet de mieux appréhender notre problème. Dans la première situation (graphique 9.2), la population des "x" est parfaitement séparée des "o" sur l'axe des abcisses. Il en est tout autrement dans le graphique 9.3 où les populations sont parfaitement confondues. Dans ces deux cas, n'importe quel test, paramétrique ou non, est capable de trouver la bonne réponse.

Dans la troisième configuration (graphique 9.4), il apparaît également que l'étiquette "x" est parfaitement discriminable de l'étiquette "o", la seule restriction par rapport à la première situa-

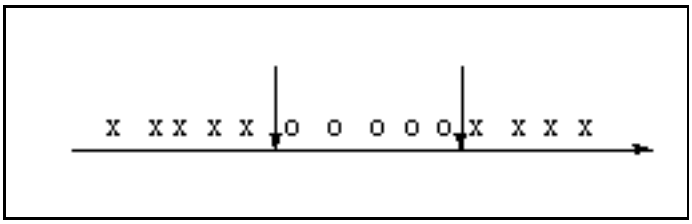

FIG. 9.4 – *Les "x" et les "o" sont-ils quand même séparables?*

tion est que la distribution des "x" est multimodale. Ici, on constate à quel point il est important de procéder à un test sur les paramètres d'échelle avant de spécifier un test d'homogénéité des populations fondé sur des paramètres de localisation. Sans cela, un test d'égalité des moyennes de Student, qui est quand même assez robuste, en concluant à une égalité des moyennes, ferait croire que la distribution des "x" est la même que celle des "o", donc à l'inopportunité de la discrétisation, ce qui est complètement erroné en l'occurrence.

Dans le cadre des exigences que nous avons formulées ci-dessus, et au vu des graphiques (9.2, et 9.4). Il existe une famille de tests qui semble bien adaptée à notre problème : les tests fondés sur la notion de séquences.

Test des séquences de Mood

Wald et Wolfowitz d'abord [Wald et Wolfowitz, 1940] pour les problèmes à deux classes, Mood par la suite généralisant à un nombre d'étiquettes supérieur à deux [Mood, 1940], ont proposé un test d'homogénéité des populations fondées sur le *nombre de séquences observées*. Notons que dans leurs travaux, il ne considéraient pas le cas d'ex-aequo, la probabilité que deux individus prennent la même valeur sur $X(.)$ est nulle dans $\mathbb{R}$. Dans la pratique, cette condition est rarement vérifiée, les mesures, même si elles s'appliquent à des concepts continus, sont souvent discrètes ordinales (on mesure la taille en centimètres, on ne se préoccupe pas des fractions en millimètres, microns...). Une manière très simple de contourner cet écueil est d'adopter la méthode des rangs aléatoires : lorsque deux individus ont la même valeur, on décide que l'un est plus grand que l'autre de manière aléatoire, ainsi toutes les propriétés des tests non-paramètriques restent valables [Caperaa et Cutsem, 1988].

Posons $e_k = \frac{n_k}{n}$, la proportion d'individus portant l'étiquette y_k. Mood a démontré que sous l'hypothèse d'homogénéité des fonctions de répartition, la quantité :

$$\frac{r - n(1 - \sum_{k=1}^{K} e_k^2)}{\sqrt{n}}$$

suit asymptotiquement une loi normale de moyenne nulle et de variance

$$\sigma^2 = \sum_{k=1}^{K} e_k^2 - 2\sum_{k=1}^{K} e_k^3 + (\sum_{k=1}^{K} e_k^2)^2$$

Le test est **unilatéral à gauche**, la région critique de rejet de l'hypothèse nulle pour un niveau de confiance $1-\alpha$ s'écrit :

$$r < n(1-\sum_{k=1}^{K} e_k^2) - U_{1-\alpha}\sqrt{n}\sqrt{\sum_{k=1}^{K} e_k^2 - 2\sum_{k=1}^{K} e_k^3 + (\sum_{k=1}^{K} e_k^2)^2}$$

où $U_{1-\alpha}$ est la valeur critique lue dans la table de la loi normale centrée et réduite. Notons que pour les petites valeurs de n, la distribution théorique des séquences a été tabulée par [Rakotomalala, 1995b], largement reprise dans [Rabaseda, 1996]. Il apparaît nettement dans ces études que la distribution asymptotique soit assez robuste.

Le test des séquences de O'Brien

Le test de Mood est applicable dans une très grande variété de problèmes. Néanmoins, étant peu spécifique, il est également peu puissant. Un autre chercheur s'est penché sur un deuxième test toujours fondé sur la notion de séquences, il repose dans ce cas sur la *longueur des séquences*, et a été élaboré par [O'Brien et Dyck, 1985]. L'hypothèse nulle est l'apparition aléatoire d'une observation de la classe y_k le long de l'axe des abcisses formé par un attribut prédictif $X(.)$, c'est-à-dire "toutes les permutations sont équiprobables". Notons que ce test se différencie du précédent dans la mesure où il qualifie de non-aléatoire la discrimination parfaite comme le mélange total.

Pour fixer les idées, nous considérons le cas particulier du problème à deux classes : les "x" et les "o". Observons l'exemple suivant :

$$xxoxoooooxxxoo$$

Le nombre de séquences est bien égale à $r = 6$, les séries de x et de o sont respectivement de longueur $2-1-3$ et $1-4-2$.

La statistique du test repose sur une combinaison linéaire de la variance de la longueur des séquences des différentes étiquettes. Soient :

- n_k l'effectif des individus portant la classe k,
- r_k le nombre de séquences pour l'étiquette k,
- s_k^2 la variance de la longueur des séquences des individus de la classe k,
- $c_k = \frac{(r_k^2-1)(r_k+2)(r_k+3)}{2r_k(n_k-r_k-1)(n_k+1)}$,
- $v_k = \frac{c_k\, n_k(n_k-r_k)}{r_k(r_k+1)}$.

La statistique proposée par [O'Brien et Dyck, 1985] s'écrit :

$$\chi_f^2 = \sum_{k=1}^{K} c_k s_k^2$$

et suit asymptotiquement une loi de χ^2 avec $f = \sum_{k=1}^{K} v_k$ degrés de liberté. Nous rejetons l'hypothèse de distribution aléatoire des étiquettes sur $X(.)$ si, pour un risque critique $1 - \alpha$:

$$P(\chi^2 > \chi^2_f) > 1 - \alpha$$

L'analyse menée par les auteurs montre que ce test est nettement plus puissant que celui fondé sur le nombre des séquences. En revanche, il n'est pas opérationnel pour juger directement de la séparabilité des individus. En effet, la variance du nombre de séquences pour une étiquette k est nulle dans les deux cas antagonistes suivants :

- il n'y a qu'une seule séquence de la classe k,
- les étiquettes sont alternées le long de l'axe de la variable $X(.)$, toutes les séquences de l'étiquette k sont de longueur 1.

Dans l'optique de la discrétisation, le premier cas permet de conclure à la séparabilité et d'enclencher la procédure de discrétisation. Dans le second cas par contre, le mélange parfait, même s'il est suspect du point de vue de l'indépendance des distributions, ne permet pas de déduire des bornes de discrétisation qui nous rapprocheraient de notre objectif initial..

De notre point de vue, il semble préférable de coupler les deux tests, d'autant plus que reposant sur les mêmes structures de données, ils peuvent mis en oeuvre simultanément : on trie d'abord les observations selon les valeurs de $X(.)$, puis on compte les séquences et leurs longueurs dans une même passe. Dès lors, nous utiliserons tout d'abord le test de Mood, qui a tendance à trop souvent conclure à la séparabilité, il permet d'exclure d'office tous les attributs qui ne présentent aucun intérêt du point de vue de la discrétisation. Le test d'O'Brien sélectionne ensuite les meilleures variables.

Discussions sur l'opportunité du test de séparabilité dans la discrétisation

Les tests statistiques que nous avons présentés ci-dessus possèdent l'avantage appréciable de reposer sur des formulations statistiques bien connues, leur utilisation comme préalable à la sélection des attributs candidats à la discrétisation semble naturelle [Rabaseda *et al.*, 1995]. On peut se poser néanmoins la question de savoir si elle est pertinente dans le cadre très particulier de l'apprentissage automatique. En effet, en terme de complexité de calcul, le tri des observations est commun aux deux procédures (test des séquences et discrétisation), par la suite si l'on s'en tient aux procédures les plus simples de discrétisation en L intervalles [Kerber, 1992] que nous étudierons plus loin, le surcoût de la fusion des séquences successives est faible par rapport à leur comptage et au calcul de leur longueur. Dès lors, si l'on dispose de mesures de qualité de partitions qui permettent de confronter des découpages en nombre d'intervalles différents, lorsque la discrétisation aboutit à la constitution d'un seul intervalle, on décidera qu'il n'y pas

Effectifs	Normale	Cauchy	Exponentielle
40	2.78	4.49	5.65
60	2.45	4.52	4.89
80	2.05	3.69	4.34
100	1.88	3.26	4.15
120	1.59	3.87	3.76
140	1.45	3.30	3.24
160	1.39	2.80	3.27
180	1.24	2.78	3.29
200	1.18	2.40	2.66

TAB. 9.1 – *Rapport des puissances estimées des tests sur différents effectifs et distribution des données*

de découpage pertinent du domaine de définition de l'attribut prédictif $X(.)$, et que finalement les individus n'étaient pas séparables dans cet espace : les procédures de tests ci-dessus deviennent superflues.

Afin d'évaluer la pertinence des tests de séparabilité, nous avons mené une comparaison du test de Mood, qui est préparatoire à tout test des séquences, avec un simple découpage binaire utilisant la mesure d'écart à l'indépendance du χ^2. Le rejet de l'hypothèse d'égalité des fonctions de répartition est matérialisé par l'acceptation d'un découpage à un risque critique α fixé (0.05 le plus souvent). Le calcul des expressions analytiques de la puissance du test étant trop compliqué, nous avons préféré générer des observations à deux classes suivant plusieurs lois diverses, et à mesurer empiriquement le rapport de la puissance (qui est égal à $1 - risque\_de\_deuxieme\_espece$) [46] du découpage χ^2 face au test de Mood à des tailles d'échantillons différentes. Les distributions que nous avons générées sont inspirées des travaux de [Caperaa et Cutsem, 1988], elles couvrent une large classe de problèmes :

1. une loi normale d'écart-type 1, de moyenne 0 (resp. 1) pour la classe x (resp. o);
2. une loi de Cauchy de paramètre 0 (resp. 1) pour la classe x (resp. o);
3. une loi exponentielle de paramètre 1 (resp. 2) pour la classe x (resp. o);

Les résultats consignés dans le tableau 9.1 montrent de manière édifiante qu'un simple test du χ^2 sur un tableau de contingence surclasse la procédure statistique de Mood. L'exploration des données à l'aide de ce test ne se justifie pas dans le cadre d'une phase préparatoire à la discrétisation supervisée des variables continues.

46. la probabilité d'accepter l'hypothèse d'homogénéité sachant que l'hypothèse alternative est vraie

9.4 Choix du type de la discrétisation : le débat supervisé - non-supervisé

La méthode de discrétisation la plus évidente est la discrétisation manuelle : un expert du domaine utilise ses propres connaissances pour déterminer les bornes de discrétisation les plus pertinentes compte tenu ou non des relations de la variable continue avec la variable d'intérêt que l'on cherche à prédire. Par exemple, on peut découper l'âge en deux intervalles simples qui sont "mineur" ($< 18ans$) et "majeur" ($\geq 18ans$). Si le sujet d'étude est le chômage, l'âge pourra être subdivisé en trois intervalles ($\leq 20ans$) et ($\geq 60ans$) désignent les "inactifs", ($20 < \quad et \quad < 60$) désigne les "actifs".

En apprentissage automatique, il est nécessaire de trouver des heuristiques qui assurent un découpage pertinent, compte tenu de l'objectif que nous avons formulé dans 9.2.1, dans ce contexte il paraît difficile de procéder au découpage du domaine de l'attribut prédictif, "en aveugle", sans tenir compte de la classe à prédire. D'ailleurs les méthodes les plus connues en apprentissage automatique ne tiennent compte que des informations apportées par la distribution des classes des individus. Les mesures de qualité des partitions sont appliquées sur le tableau de contingence issu du découpage en intervalles disjoints. Certains auteurs [Dougherty *et al.*, 1995] affirment que dans le cadre de l'apprentissage supervisé, l'utilisation du gain d'entropie est pertinente dans la recherche des meilleurs points de discrétisation, même s'il est nécessaire de disposer d'un grand nombre d'individus pour obtenir une estimation fiable [de Merckt, 1993] [Wehenkel, 1997]. Malgré tout, il existe quelques méthodes qui se proposent d'effectuer cette opération en ignorant la distribution des classes des individus, ou en les introduisant comme guide dans un processus ad-hoc. Dans cette section, nous nous concentrerons sur les méthodes non-supervisées en le positionnant face aux méthodes supervisées.

9.4.1 Les insuffisances des méthodes "traditionnelles" de découpage

Avant le récent développement des méthodes de discrétisation ces dernières années, les méthodes les plus couramment utilisées étaient le découpage en intervalles d'effectifs ou de largeurs égales, le nombre d'intervalles étant fixé arbitrairement à partir de considérations empiriques ou en utilisant la formule $L = \log n + 1$ que l'on doit plutôt considérer comme "le minorant du nombre d'intervalles adéquat" [Aivazian *et al.*, 1986]. Cette procédure se justifiait surtout par l'élaboration de regroupements dans la perspective d'une simplification du traitement des données, le terme consacré était alors le *recodage* de variables quantitatives en variables discrètes. Elles ont été rapidement abandonnées tant les résultats obtenus étaient mauvais dans l'induction de règles [Celeux et Robert, 1993]. Les découpages ne tenant compte ni de la proximité des individus, ni de leur étiquette, il est normal que la perte d'information soit considérable. Dans

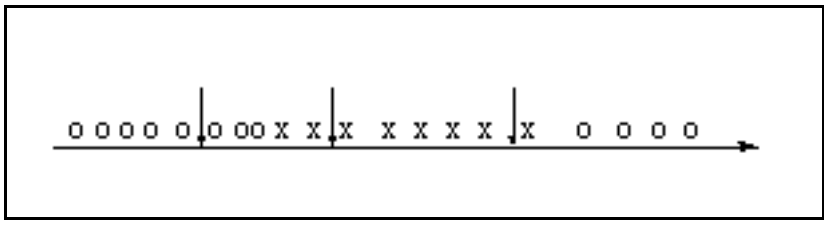

FIG. 9.5 – *La discrétisation en quatre intervalles d'effectifs égaux produit un découpage induisant une perte d'information non-contrôlée*

l'exemple de la figure 9.5, nous observons très bien qu'un découpage en 4 intervalles d'effectifs égaux produit un résultat qui laisse perplexe, en tous les cas ne nous rapprochant en rien de notre objectif initial.

Certes il existe bien des variantes de ces premiers algorithmes utilisant partiellement les informations supervisées. On notera par exemple les méthodes qui visent à ajuster les bornes précédentes en minimisant l'entropie dans chaque intervalle [Wong et Chiu, 1987]. Néanmoins, face aux méthodes purement supervisées, elles présentent peu d'intérêt.

9.4.2 Discrétisation non-contextuelle utilisant les informations de similarités entre les exemples

La question que l'on est en droit de se poser est : "Peut-on espérer un découpage intéressant si l'on ne tient pas compte explicitement de la classe à prédire?"

Il existe de nombreuses méthodes non-supervisées pour découper une variable continue, nous nous intéresserons plus particulièrement aux travaux de [de Merckt, 1993] parce qu'il les a développés dans le cadre de l'induction par graphes. Il part du constat selon lequel la discrétisation fait perdre l'information importante qu'est la proximité des individus dans l'espace de représentation constitués par l'attribut prédictif continu. De fait, nous devons essayer de trouver un découpage qui respecte au mieux la similarité au sein du groupe défini par l'intervalle. L'auteur se cantonne dans la discrétisation binaire. Soient :

- G_1 et G_2 les deux ensembles définis sur le domaine D_X en partitionnant au point de discrétisation d i.e $G_1 = \{\omega \in \Omega_a \,/\, X(\omega) \leq d\}$ et $G_2 = \{\omega \in \Omega_a \,/\, X(\omega) > d\}$
- $\overline{x}_i = \frac{1}{|G_i|} \sum_{\omega \in G_i} X(\omega)$

Il propose la mesure "Contrast" suivante :

$$Contrast(G_1, G_2, d) = \frac{|G_1| \,.\, |G_2|}{|G_1 \cup G_2|} (\overline{x}_1 - \overline{x}_2)$$

qui s'avère être tout simplement l'inertie inter-groupes monovariée. Le découpage revient

donc à construire deux groupes qui minimisent la distance entre les individus à l'intérieur des groupes, et maximisent la distance entre les centres de gravité des groupes.

Cette idée n'est pas nouvelle. [Fischer, 1958], déjà en 1958, a proposé un algorithme optimal de partitionnement d'une variable continue en L intervalles, il est inspiré de la programmation dynamique et est de complexité quadratique [Diday *et al.*, 1982]. La mesure utilisée est la maximisation de l'inertie inter-groupes ou, ce qui revient au même, la minimisation de l'inertie intra-groupes.

Malgré les qualités indéniables de cet algorithme, et bien que [de Merckt, 1993] ait testé avec succès sa méthode, il est vrai sur deux bases de données, dont une, celle des Iris, réputée très facilement séparable sur un des attributs prédictifs, il reste assez vulnérable et peut parfois engendrer des découpages assez peu intéressants [Celeux et Robert, 1993]. Plus prometteurs semble-t-il est l'utilisation conjointe des informations apportées par les classes et les similarités lors de la recherche de la borne de discrétisation.

9.4.3 Complémentarité des approches supervisées - non-supervisées

Les attributs continus possèdent un avantage intéressant sur les variables catégorielles en ce sens que l'on peut calculer une mesure de proximité entre les individus. L'argument, très sensé, de quelques auteurs est de mélanger les deux approches afin de bénéficier du maximum d'informations. [Chmielevski et Grzymala-Busse, 1994] par exemple proposent d'effectuer tout d'abord une classification en L intervalles avant de corriger les bornes en utilisant une heuristique proche de celle d'Anderberg (1974) [Diday *et al.*, 1982]. L'objectif dans cette deuxième étape est de minimiser l'entropie de la partition. D'autres encore [Chan *et al.*, 1991] proposent de construire des intervalles binaires de longueurs identiques, puis de construire un arbre d'induction à l'aide d'ID3, de détecter alors les intervalles responsables d'erreurs et de les découper à nouveau.

Le vrai problème réside en fait dans l'élaboration d'une mesure appropriée afin d'évaluer les partitions engendrées. [de Merckt, 1993] propose de calculer le rapport entre l'inertie inter-groupes et l'entropie.

$$CE(G_1, G_2, d) = \frac{Contrast(G_1, G_2, d)}{-\frac{n_{.1}}{n}\sum_{k=1}^{K}\frac{n_{k1}}{n_{.1}}\log(\frac{n_{k1}}{n_{.1}}) - \frac{n_{.2}}{n}\sum_{k=1}^{K}\frac{n_{k2}}{n_{.2}}\log(\frac{n_{k2}}{n_{.2}})}$$

Son indicateur favorise les découpages "très contrastés" (les deux groupes sont éloignés l'un de l'autre) et avec une faible entropie (il y a peu de mélange des classes dans chaque groupe). Dans la pratique, lorsque les partitions sont pures, sans mélanges de classes, la quantité au dénominateur qui est naturellement nulle est remplacée arbitrairement par la valeur 10^{-6}.

Cette dernière remarque montre à quel point il est difficile de construire un indicateur fondé sur des approches théoriques solides dans l'état actuel de la recherche. On voit effectivement, surtout si l'on utilise l'interprétation de l'entropie en terme de variance, que le rapport est proche

celui de l'analyse de variance à un facteur. Au numérateur, nous avons bien la variance inter-classes, expliquée, et au dénominateur, la variance résiduelle, mais calculée sur une représentation différente, à savoir les classes. Si l'interprétation est viable, l'expréssion manque d'assise théorique, et dans ce cadre on peut se demander si une approche en termes d'inertie ne serait pas meilleure. En effet, partant toujours du constat que l'entropie peut être interprétée en variance, on gagnerait vraisemblablement en clarté si l'on recombinait l'expression de manière à obtenir une inertie, qui n'est autre que la somme pondérée des variances intra-classes. Deux écueils pour l'instant nous empêche d'aller plus avant dans cette direction : la première est purement pragmatique, tous nos tests montre que cette approche ne se révèle pas significativement meilleure que la méthode *Contraste*; la seconde est autrement plus ennuyeuse, les variables n'ont forcément pas la même variance au départ, surtout s'agissant de variables numériques et nominales, certaines pourraient alors influer inconsidérément sur les résultats. Certes, nous pourrions adopter une solution radicale en réduisant toutes les variables mais nous penchons vers des stratégies plus élaborées sous forme de pondération. A l'heure actuelle, nous ne disposons pas de résultats probants sur ce problème.

9.5 La discrétisation supervisée en L intervalles

Ce sujet est certainement celui qui a le plus connu de développements ces dernières années. La cause principale était qu'elle offrait de nouvelles perspectives à des algorithmes symboliques jusque là cantonnés aux données de type qualitatif. Les méthodes les plus souvent citées dans les papiers d'origine anglo-saxonne sont celles de [Fayyad et Irani, 1993] en intelligence artificielle, et [Kerber, 1992] en approche statistique. Chacune d'elles répond à une logique propre : la première effectue des partitions binaires récursivement jusqu'à ce qu'une condition d'arrêt soit vérifiée, la seconde part au contraire de la partition la plus fine, un individu dans chaque intervalle, avant de les rassembler jusqu'à ce qu'une condition d'arrêt soit vérifiée. Assez curieusement, ces méthodes reposent sur des mesures de qualité de partitions locales. Or, sachant que l'on peut disposer d'indicateurs tenant compte de la complexité du modèle (ici le nombre d'intervalles), ou ce qui revient au même comme nous avons pu le constater dans le chapitre précédent, tenant compte explicitement de la taille des effectifs dans chaque intervalle, il paraît plus avantageux de trouver un découpage qui optimise globalement cette mesure, la sélection du nombre L se fera alors en comparant les valeurs de l'indicateur sur les partitions optimales de tailles différentes.

En France, [Lechevallier, 1990] a beaucoup travaillé sur l'extension de l'algorithme de [Fischer, 1958] dans le cadre supervisé en utilisant différentes mesures, toutes additives, condition sine qua non de son application. Sa stratégie assure la découverte de la partition optimale en $O(n^2)$. Il existe outre-atlantique quelques tentatives de discrétisation optimale utilisant toujours la programmation dynamique mais en méconnaissant complètement les résultats de Fischer et Lechevallier.

Ces algorithmes restent assez limités, soit par la mesure utilisée (le taux d'erreur en resubstitution pour [Maas, 1994], utilisé dans la méthode d'induction T2 [Auer *et al.*, 1995]), soit par le nombre de classes qu'elles peuvent prendre en compte (problème à deux classes uniquement pour [Fulton *et al.*, 1995]).

Dans la section suivante, nous présentons un nouvel algorithme de discrétisation qui s'inspire de la stratégie de Lechevallier mais dont la complexité est moindre car on effectue au préalable une sélection judicieuse des points de discrétisation candidats.

9.5.1 Discrétisation optimale

Condition d'application de l'algorithme de Fischer

L'algorithme de Fischer repose sur la programmation dynamique, [Lechevallier, 1990] a démontré que sous les conditions suivantes, la méthode était bien optimale :

- **Propriété d'ordre** : sur l'ensemble $X(\Omega) = \{x_1, ..., x_n\}$ une partition en L intervalles possède la propriété d'ordre si pour deux éléments x_i et x_j de $X(\Omega)$ classés dans le même intervalle I_l, toute valeur comprise entre x_i et x_j appartient au même intervalle, c'est à dire :

  $$\forall x_i, x_j \in X(\Omega) \ , \ (x_i < x_j \, et x_i \in I_l \, et x_j \in I_l \,) \Rightarrow (\forall x_k \in X(\Omega) \ / \ x_i < x_k < x_j \Rightarrow x_k \in I_l)$$

- **Propriété d'additivité de la mesure de qualité :** si la partition décrite par le découpage $(\{x_1, ..., x_i\}, I_2, ..., I_L)$ est optimale pour L intervalles, alors la partition décrite par le découpage $(I_2, ..., I_L)$ est une partition optimale en $L-1$ intervalles de $\{x_{i+1}, ..., x_n\}$.

La première propriété ne pose guère de problème, sachant que $X(\Omega) \subseteq \mathbb{R}$ nos observations sont nécessairement régies par une structure d'ordre. En revanche, la seconde requiert l'additivité de la mesure de qualité de la partition utilisée. Cela a été démontré par Lechevallier [Lechevallier, 1990] pour les mesures PRE ou issues du χ^2, [Zighed et Rakotomalala, 1996a] pour la mesure $\varphi(T)$.

Choisir une "bonne" mesure d'évaluation des partitions

Nous sommes nécessairement amenés à comparer des partitions de Ω induites par un découpage en un nombre d'intervalles différents. Les mesures de qualité que nous utilisons doivent tenir compte explicitement de l'accroissement de complexité résultant d'un découpage excessif. En effet, on sait qu'un découpage très fin, par exemple un individu dans chaque intervalle, présentera un taux d'erreur en resubstitution quasiment nul sur l'échantillon d'apprentissage [Maas, 1994], en revanche il est à prévoir que le modèle proposé sera peu fiable. En tous les cas,

comme nous avons pu le constater dans le chapitre précédent, le taux d'erreur est inadapté en apprentissage car il est incapable de traiter convenablement les problèmes où les classes ne sont pas parfaitement équidistribuées.

L'introduction d'un biais de complexité lors de l'évaluation de la partition proposée exclut d'office les mesures fondées exclusivement sur une estimation des probabilités à partir des fréquences empiriques. Par exemple, les mesures de pureté ou d'autres comme le gain informationnel présentent la particularité de favoriser systématiquement un découpage excessif des domaines de $X(.)$. En fait, les mesures les plus adaptées ici sont celles qui pénalisent : les découpages trop complexes, matérialisés par un grand nombre d'intervalles; ou encore qui produisent des intervalles de trop petite taille en effectifs.

Dans notre travail nous avons décidé d'utiliser la mesure de [Zighed *et al.*, 1996], elle est simple à calculer, additive, et répond parfaitement aux exigences posées ci-dessus. Nous verrons de plus qu'elle possède une qualité intéressante qui nous permet de réduire l'espace de recherche.

Réduction de la complexité de résolution par un choix approprié des bornes de discrétisation candidats

La recherche de la partition optimale en L intervalles sur un échantillon de n points peut être faite de manière directe en explorant toutes les solutions possibles. [de Merckt et Quinlan, 1996] l'ont d'ailleurs expérimentée pour un découpage en trois intervalles. Dans ce cas, l'algorithme possède une forte complexité que l'on évalue à $O(n^{L-1})$. En effet, le nombre de partitions candidates est

$$\begin{pmatrix} n-1 \\ L-1 \end{pmatrix}$$

Et si l'on veut chercher le nombre d'intervalles optimal, le nombre de partitions à évaluer devient

$$\sum_{L=1}^{n} \begin{pmatrix} n-1 \\ L-1 \end{pmatrix}$$

Avec la méthode de Fischer implémentée par [Lechevallier, 1990], la complexité est réduite à $O(n^2)$. On peut d'ores et déjà considérer le problème de la discrétisation résolue, du moins dans le cadre de la recherche de l'optimalité, puisque l'on peut trouver les meilleures bornes dans l'absolu en un temps de calcul raisonnable [Zighed *et al.*, 1997]. Néanmoins, ces méthodes étant destinées à appréhender des bases de données gigantesques [Catlett, 1991b], on a intérêt à ce que la discrétisation dans une phase de préparation des données permette un traitement aussi rapide que possible afin de réduire l'espace de recherche en éliminant par exemple les variables sur lesquelles le découpage a échoué [Liu et Setiono, 1996].

Dans la section 9.2.2, nous avons défini ce qu'était les séquences et les points frontières. Dans le cas d'absence de coïncidence entre les observations sur l'axe formé par la variable $X(.)$, les points

frontières n'encadreraient que des groupes purs au sens où ils ne sont constitués que d'une classe. Il apparaîtrait alors étrange qu'un algorithme de discrétisation cherchant à regrouper les individus, utilisant un critère de pureté ou d'entropie produise une borne qui n'appartienne pas à l'ensemble des points frontières détectés sur l'échantillon. [Fayyad et Irani, 1992] ont d'ailleurs démontré qu'un point de discrétisation optimal doit obligatoirement appartenir aux points frontières si l'on utilise une mesure d'entropie. Nous allons montrer ici que ce théorème peut être étendu à la mesure sensible à la taille des effectifs $\varphi(T)$ de [Zighed *et al.*, 1996].

Theorem 11 *Si d est un point de discrétisation minimisant la mesure $\varphi(T)$ alors d est nécessairement un point frontière.*

Preuve : Rappelons que la mesure appliquée sur un tableau de contingence T représentant une partition en L intervalles s'écrit

$$\varphi(T) = \sum_{l=1}^{L} \alpha \frac{n_{.l}}{n} \sum_{k=1}^{K} \frac{n_{kl}+\lambda}{n_{.l}+K\lambda}(1 - \frac{n_{kl}+\lambda}{n_{.l}+K\lambda}) + (1-\alpha)\frac{K\lambda}{n_{.l}}$$

Pour simplifier notre exposé, et sans restreindre pour autant la portée de notre discours, nous nous placerons dans le cas d'un découpage binaire avec un problème à deux classes. La situation que nous étudions est la suivante (figure 9.6)[Fayyad et Irani, 1992]. On cherche à démontrer que $\varphi(T)$ est optimal si et seulement si n_{11} est tel que :

- $n_{11} = 0$
- $n_{11} = n_{1.}$

Dans les deux cas, d est un point frontière : soit d_1, soit d_2.

Sur notre exemple, la mesure $\varphi(T)$ s'écrit :

$$\begin{aligned}
\varphi(T) \quad = \quad & [\alpha\frac{(n_{11}+n_{21})}{n}(\frac{n_{11}+\lambda}{n_{11}+n_{21+2\lambda}}(1-\frac{n_{11}+\lambda}{n_{11}+n_{21+2\lambda}}) + \frac{n_{21}+\lambda}{n_{11}+n_{21+2\lambda}}(1-\frac{n_{21}+\lambda}{n_{11}+n_{21+2\lambda}})) \\
& +(1-\alpha)\frac{2\lambda}{(n_{11}+n_{21})}] \\
& +[\alpha\frac{(n_{1.}-n_{11}+n_{22})}{n}(\frac{n_{1.}-n_{11}+\lambda}{n_{1.}-n_{11}+n_{22}+2\lambda}(1-\frac{n_{1.}-n_{11}+\lambda}{n_{1.}-n_{11}+n_{22}+2\lambda}) \\
& +\frac{n_{22}+\lambda}{n_{1.}-n_{11}+n_{22}+2\lambda}(1-\frac{n_{22}+\lambda}{n_{1.}-n_{11}+n_{22}+2\lambda})) + (1-\alpha)\frac{2\lambda}{(n_{1.}-n_{11}+n_{22})}]
\end{aligned}$$

Après de nombreuses tentatives, il est apparu qu'il était très difficile d'extraire une solution littérale de la minimisation de $\varphi(T)$ pour n_{11} variant entre 0 et $n_{1.}$. Nous avons donc adopté une autre démarche : montrer qu'il existe un maximum unique avec une dérivée seconde toujours négative dans cette région.

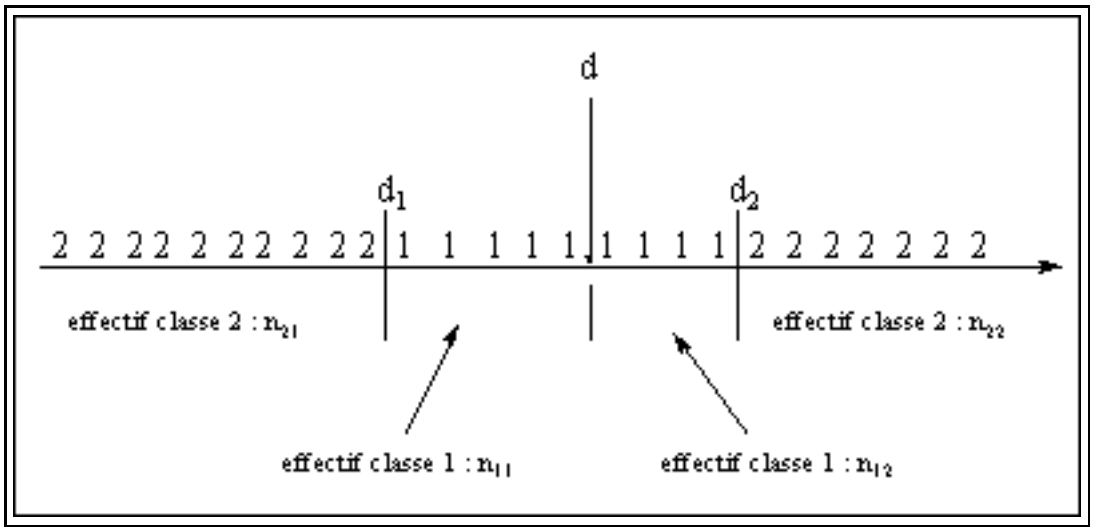

FIG. 9.6 – *Le point de discrétisation d peut-il être à un autre emplacement que d_1 et d_2 ?*

La preuve est générale mais afin d'alléger les notations, nous avons fixé $\lambda = 0$ et $\alpha = 1$. La dérivée première de $\varphi(T)$ s'écrit

$$\frac{\partial \varphi(T)}{\partial n_{11}} = 2\frac{n_{21}^2 n_{1.}^2(1 + 2n_{22} - 2n_{11}) - n_{21}n_{11}(2n_{22}^2 + n_{21}n_{11} - 2n_{21}n_{22} - n_{11}n_{22}^2/n_{21})}{(n_{1.} - n_{11} + n_{22})^2(n_{11} + n_{21})^2(n_{1.} + n_{21} + n_{22})}$$

elle comporte deux racines réelles

$$r_1 = -\frac{n_{21}(2n_{22} + n_{1.})}{n_{22} - n_{21}}$$

qui peut être négative ou en-dehors de l'intervalle $[0, n_{1.}]$, elle ne nous intéresse pas ici, et

$$r_2 = \frac{n_{21}n_{1.}}{n_{21} + n_{22}}$$

qui est compris entre 0 et $n_{1.}$. Et on montre assez facilement que la fonction $\varphi''(T)$, dérivée seconde $\frac{\partial^2 \varphi(T)}{\partial^2 n_{11}}$ de $\varphi(T)$, est toujours négative sur cet intervalle.

Ce résultat nous permet de réduire considérable l'espace de recherche, l'algorithme de Fischer passe à une complexité $O(u^2)$, ceci étant très avantageux si u est suffisamment petit face à n. Afin de nous en assurer, nous allons nous placer dans le pire des cas, celui où la distribution des classes sur les individus est aléatoire. Cette situation n'est pas sans rappeler l'hypothèse nulle du test de Mood [Mood, 1940], le nombre de séquences est alors en moyenne de $E(r) = n(1 - \sum_{k=1}^{K}(\frac{n_k}{n})^2)$ où n_k désigne la proportion d'individus de la classe k. Sachant qu'il y a une relation directe $u = r - 1$, nous pouvons donc quantifier le nombre moyen de points frontières dans le cas où la disposition des points est aléatoire :

$$E(u) = n(1 - \sum_{k=1}^{K}(\frac{n_k}{n})^2) - 1$$

En moyenne donc, dans le cas de la distribution aléatoire des observations, il y a une réduction du nombre de points à considérer dans la discrétisation égale à $n \times \sum_{k=1}^{K} (\frac{n_k}{n})^2$, toujours positive. Notons que cette quantité est minimale quand les classes sont équidistribuées, elle diminue avec l'augmentation du nombre de classes associées au problème. Dans les applications où certaines classes prédominent, le nombre de points frontières est faible par rapport à l'effectif de l'échantillon.

Bien entendu, ce facteur peut être considéré comme un minorant puisque dans la plupart des cas, la disposition des individus n'est pas aléatoire, notre objectif étant justement de détecter les "formes" matérialisées par la proximité des individus portant la même étiquette. Nous avons évalué empiriquement les temps de calculs comparés pour une discrétisation optimale en 4 intervalles sur les Ondes de Breiman (300 individus) en utilisant exactement la même implémentation, mais en désactivant le filtre permettant de constituer les points frontières par les séquences dans le second algorithme.

Les temps affichés (Table 9.2) sont en millisecondes sur un Pentium 90Mhz. On constate que le temps de calcul est très stable lorsque l'on teste tous les points de discrétisation situés entre chaque individu (299 points). En revanche, la détection des points frontières permet de réduire considérablement les temps de calculs lorsque les observations sont séparables (variable v9), et de toute manière les temps n'excèdent jamais 7 secondes. Nous remarquons que sur toutes les variables, les bornes de discrétisation produites ont toujours été les mêmes, ce qui confirme si besoin était qu'elles sont effectivement parmi les points frontières. En moyenne, cette nouvelle stratégie est 10 fois plus rapide (sur ce fichier) que l'ancienne méthode.

Algorithme optimal de discrétisation

Considérons la partition en l éléments de Ω définie comme suit

$$\mathcal{P}_l^1 = \{\{R_1 \cup \ldots \cup R_{j_1}\}, \{R_{j_1+1} \cup \ldots \cup R_{j_2}\}, \ldots, \{R_{j_{l-1}+1} \cup \ldots \cup R_r\}\}$$

à laquelle nous associons le tableau de contingence T. Dans ce qui suit, nous noterons indifféremment $\varphi(T)$ ou $\varphi(\{\{R_1, \ldots, R_{j_1}\}, \{R_{j_1+1}, \ldots, R_{j_2}\}, \ldots, \{R_{j_{l-1}+1}, \ldots, R_r\}\})$ la mesure de qualité de la partition $\mathcal{P}_l^1$.

Nous considérons donc une séquence $\{R_i, 1 \leq i \leq r\}$ et nous cherchons une partition sur Ω ordonnée optimale pour la mesure φ. La partition en k classes cherchée, notée $\mathcal{P}_l^1$, est de la forme :

$$\mathcal{P}_l^1 = \{\{R_1, \ldots, R_{j_1}\}, \{R_{j_1+1}, \ldots, R_{j_2}\}, \ldots, \{R_{j_{l-1}+1}, \ldots, R_r\}\}$$

La mesure φ étant additive, la valeur de φ sur le tableau T induite par la partition précédente est donnée par $\sum_{i=0}^{l} \varphi(\{R_{j_{i-1}+1}, \ldots, R_{j_i}\})$ où $j_0 = 1$ et $j_l = r$ pour plus de simplicité. L'additivité

Variable	Points fontières	Tous les points	Rapport
v1	6261	46695	7.5
v2	5119	46736	9.1
v3	4551	46732	10.3
v4	5030	46735	9.3
v5	5860	46749	8.0
v6	5868	49714	8.0
v7	5019	46761	9.3
v8	5744	46704	8.1
v9	3716	46762	12.6
v10	4447	46733	10.5
v11	4344	46767	10.8
v12	5910	46710	7.9
v13	5526	46899	8.5
v14	6098	46861	7.7
v15	6084	46807	7.7
v16	3636	46674	12.8
v17	3810	46782	12.3
v18	5654	46705	8.3
v19	4168	46762	11.2
v20	5450	46683	8.6
v21	7062	46736	6.6

TAB. 9.2 – *Temps comparés (en millisecondes sur un Pentium 90 Mhz) : algorithme de Fischer, avec et sans constitution des séquences*

de la mesure implique que la partition

$$\mathcal{P}_{l-1}^{j_1+1} = \{\{R_{j_1+1}, \ldots, R_{j_2}\}, \ldots, \{R_{j_{l-1}+1}, \ldots, R_r\}\}$$

est une partition optimale en $l-1$ intervalles de l'ensemble $\{R_i, j_1+1 \leq i \leq r\}$. Déterminer la partition optimale en l intervalles est donc équivalent à rechercher le premier point de coupe j_1. Ce point peut prendre n'importe laquelle des valeurs de l'ensemble discret $\{1, ..., r-l+1\}$. La partition optimale $\mathcal{P}_l^1$ peut ainsi être définie par un problème de minimisation :

$$\varphi\left(\mathcal{P}_l^1\right) = \min_{1 \leq j_1 \leq r-l+1} \left\{\varphi\left(\{R_1, \ldots, R_{j_1}\}\right) + \varphi\left(\mathcal{P}_{l-1}^{j_1+1}\right)\right\}$$

Si nous savons déterminer les différentes partitions $\mathcal{P}_{l-1}^{j_1+1}$, alors nous savons par le procédé de minimisation précédent déterminer $\mathcal{P}_l^1$. Il nous faut maintenant construire les partitions $\mathcal{P}_{l-1}^{j_1+1}$. Pour cela, il suffit de remarquer que vis-à-vis de l'ensemble de séquences $\{R_i, j_1+1 \leq i \leq r\}$, $\mathcal{P}_{l-1}^{j_1+1}$ joue exactement le même rôle que $\mathcal{P}_l^1$ pour l'ensemble $\{R_i, 1 \leq i \leq r\}$. Ainsi, le procédé précédent peut être à nouveau utilisé. Ceci nous conduit donc à déterminer les partitions $\mathcal{P}_{l-2}^{j_1'+1}$ qui seront les partitions optimales en $l-2$ intervalles de la partie droite des partitions ordonnées de $\{R_i, j_1+1 \leq i \leq r\}$. On peut ainsi de proche en proche ramener la connaissance des partitions $\mathcal{P}_{l-1}^{j_1+1}$ à la connaissance des partitions $\mathcal{P}_1^k$ qui représentent les partitions optimales en un seul intervalle de l'ensemble de séquences $\{R_i, k \leq i \leq r\}$. La connaissance de ces partitions permet en utilisant la minimisation de connaître de proche en proche toutes les partitions $\mathcal{P}_p^q$ avec $1 \leq p \leq l-1$ et $1 \leq q \leq r-p+1$. Ceci conduit donc à la détermination des partitions $\mathcal{P}_{l-1}^{j_1+1}$, à partir desquelles nous déterminons enfin $\mathcal{P}_l^1$. Finalement, connaître toutes les partitions $\mathcal{P}_1^k$ permet d'obtenir la partition optimale recherchée. Mais, nous savons que $\mathcal{P}_1^k = \{\{R_k, \ldots, R_r\}\}$. Il suffit alors de calculer de proche en proche les partitions optimales intermédiaires pour obtenir la partition $\mathcal{P}_l^1$. On peut donc en déduire l'algorithme suivant :

1. Nous considérons la partition de Ω définie par la séquence $(R_1, ..., R_r)$.

2. Calculs des $\mathcal{P}_1^k$ pour $1 \leq k \leq r$

3. Pour $2 \leq p \leq r$, détermination des partitions $\mathcal{P}_p^q$ pour chaque q de l'intervalle $[1, r-p+1]$

   - Calculs des sommes $\varphi\left(\{R_q, \ldots, R_o\}\right) + \varphi\left(\mathcal{P}_{p-1}^o\right)$ pour $q \leq o \leq r$
   - $\varphi\left(\mathcal{P}_p^q\right)$ est le minimum des valeurs précédentes
     A l'issue de cette étape, nous obtenons la partition optimale $\mathcal{P}_l^1$.

4. La partition optimale $\mathcal{P}^*$ est telle que $\varphi(\mathcal{P}^*) = \min_{j=1}^r \varphi(\mathcal{P}_j^1)$

9.5.2 Les stratégies gloutonnes

L'existence d'un algorithme optimal de complexité quadratique est un résultat que l'on peut juger très satisfaisant. Mais nous devons garder à l'esprit qu'en apprentissage supervisé l'objectif n'est pas tant d'optimiser la mesure utilisée, mais plutôt de trouver la spécification qui minimise le taux d'erreur dans la population totale. Il nous paraît justifié de mesurer le gain de performances de cette optimisation dans le cadre de la discrétisation face aux stratégies gloutonnes de complexité moindre $O(u)$ qui sont les plus répandues dans la communauté de l'apprentissage automatique. Nous essayerons de répondre à cette question dans la sous-section suivante, auparavant nous présentons les méthodes de discrétisation les plus usitées.

Elles reposent sur deux stratégies gloutonnes:

- la première, dite "top-down" (TD), applique le principe "diviser pour régner". Cela consiste tout simplement à rechercher jusqu'au déclenchement de la règle d'arrêt la meilleure bipartition sur chaque sous-groupe défini récursivement. Le critère peut être alors calculé soit localement sur chaque sous-groupe à séparer [Fayyad et Irani, 1993][Catlett, 1991b], soit globalement sur la partition induite. L'ensemble U^* ($U^* \subseteq U$) est construit itérativement par ajout de points de discrétisation.

- la seconde, dite "bottom-up" (BU) utilise le principe inverse. On définit le découpage initial par l'ensemble U des points frontières candidats et l'on essaye de regrouper les intervalles adjacents jusqu'à l'optimisation de la mesure [Zighed *et al.*, 1996] ou jusqu'à ce qu'un regroupement local ne soit plus justifié [Kerber, 1992] ou en fixant une consistance limite avec l'échantillon d'apprentissage [Liu et Setiono, 1995]. L'ensemble U^* est construit itérativement mais par suppression de points parmi les points candidats.

Ces deux algorithmes très rapides présentent l'inconvénient d'être irrévocables. Chaque point de discrétisation introduit dans U^* par la stratégie "Top-down" ne peut plus être remis en cause; chaque point extrait de U^* ne peut plus être réintroduit dans la seconde. Nous allons illustrer chaque stratégie par les méthodes les mieux connues en apprentissage automatique.

Découpage binaire récursif minimisant l'entropie

La solution la plus simple consisterait à appliquer récursivement un algorithme de construction d'arbre, qui chercherait à chaque fois la meilleure partition binaire, jusqu'au déclenchement de la règle d'arrêt. [Kohavi et Sahami, 1996] l'ont d'ailleurs testé avec succès sur des bases de données réelles. Pourtant, la méthode qui a reçu le plus de suffrages est l'algorithme MDLPC de [Fayyad et Irani, 1993], peut-être parce qu'elle repose sur une formulation très élégante [de Merckt et Quinlan, 1996], la description minimale des données, qui permet de contrôler la complexité de la partition finale.

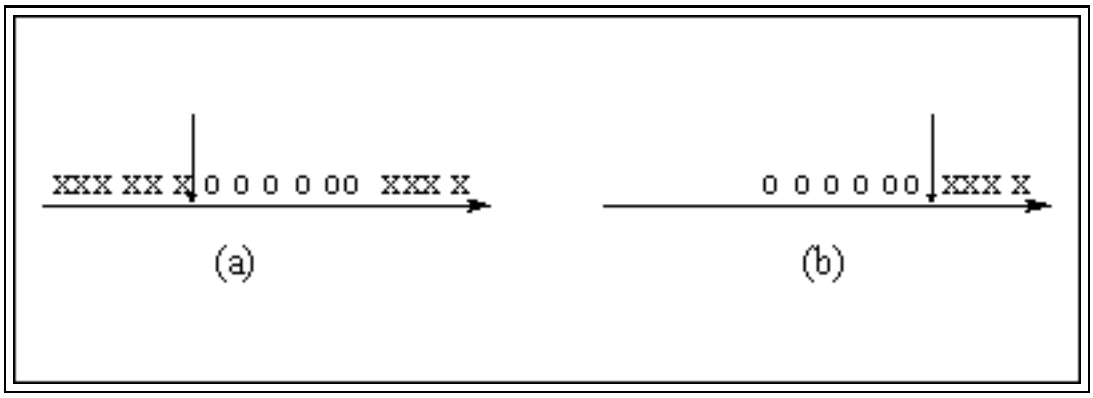

FIG. 9.7 – *Séquence de découpage par la méthode MDLPC*

MDLPC est fondé sur la mesure d'entropie de Shannon, elle recherche récursivement la partition binaire qui minimise la mesure. Dans l'exemple de la figure 9.7, elle procéderait d'abord à un premier découpage en (a) avant d'opérer, encore une fois sur les individus du second intervalle (b). Bien entendu, il n'était plus nécessaire d'essayer de découper l'intervalle situé à gauche de la première borne de discrétisation puisqu'il était composé d'individus portant la même étiquette. Ceci peut constituer une règle d'arrêt [Catlett, 1991b], les auteurs y ont joint d'autres considérations fondées sur la description minimale des messages. Le principe est simple : dans un premier temps, un émetteur décrit à un récipiendaire la classe de chaque individu composant l'échantillon d'apprentissage; on se demande si en lui envoyant une théorie sur les individus qu'il va décrire (dans notre cas, ce serait la borne de discrétisation qui selon que l'on soit à sa gauche ou à sa droite, permet de reconnaître telle ou telle classe) et les exceptions à cette théorie, il ne pourrait pas diminuer de manière drastique le coût total du message[47].

Soit un découpage en deux groupes, G_1 et G_2 :

- on note $Ent(G)$ l'entropie sur l'ensemble des individus, avec $Ent(G) = -\sum_{k=1}^{K} \frac{n_{k.}}{n} \log(\frac{n_{k.}}{n})$,
- on note $Ent(G_l)$ l'entropie sur le groupe G_l, avec $Ent(G_l) = -\sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}} \log(\frac{n_{kl}}{n_{.l}})$,
- K_l désigne le nombre de classes présentes dans le groupe G_l,
- le gain d'entropie s'écrit

$$Gain(G1, G2, d) = Ent(G) - [\frac{n_{.1}}{n} Ent(G_1) + \frac{n_{.2}}{n} Ent(G_2)]$$

- le découpage est stoppé si et seulement si, pour un point d maximisant le gain, la condition suivante est vérifiée

$$Gain(G_1, G_2, d) < \frac{\log(n-1)}{n} + \frac{\log(3^K - 2) - [K.Ent(G) - K_1.Ent(G_1) - K_2.Ent(G_2)]}{n} \quad (9.1)$$

47. Nous étudierons plus en profondeur cette théorie dans le chapitre sur la limitation de la taille des graphes

Sans entrer dans les arcanes de la théorie de la description minimale des données, l'esprit de ce critère est de contrebalancer la diminution du coût de transmission de la classe de chaque individu (avant et après découpage, le premier membre de l'équation 9.1) par le coût de la transmission de la valeur du point de discrétisation (il y a $n-1$ points possibles) et la diminution de l'encodage des classes sachant que l'on n'a pas forcément les mêmes occurrences dans chaque sous-groupe (le deuxième terme du deuxième membre de l'équation 9.1). Si l'on reprend notre exemple (figure 9.7), nous constatons effectivement que dans un problème à deux classes, seule la classe "x" est présente dans le premier intervalle. On peut se demander néanmoins si ce deuxième terme est vraiment pertinent, il n'y a aucune raison que l'émetteur change de dictionnaire de codage selon qu'il soit à gauche ou à droite du point de discrétisation, d'autres auteurs d'ailleurs ont négligé cette partie du critère [Quinlan, 1996].

De toute manière, aussi séduisante que soit cette formulation, nous avons constaté lors de nos expérimentations que l'usage d'un simple algorithme ID3 binaire récursif avec un critère d'arrêt utilisant un test statistique d'indépendance [Quinlan, 1986b] permet d'obtenir des résultats analogues : même choix des points de discrimination, ce qui est normal puisqu'on utilise le gain d'entropie dans les deux cas, et nombre d'intervalles identique.

Discrétisation par regroupement après constitution des séquences

Cette stratégie est à l'opposée de la précédente. De manière générale, il s'agit de constituer dans un premier temps les séquences, puis de fusionner les intervalles adjacents en utilisant soit un critère statistique d'équivalence distributionnelle, soit en optimisant une mesure globale. Les méthodes les plus connues dans cet ordre sont la méthode Chi-Merge de [Kerber, 1992], et Fusinter de [Zighed *et al.*, 1996]. Ces algorithmes sont décrits en détail dans [Rabaseda, 1996].

Pour notre part, dans cette section nous présenterons une contribution originale de [Liu et Setiono, 1995], nommée Chi2, qui est fortement inspirée de Chi-Merge mais en imposant une contrainte de consistance aux données, exprimée à l'aide du taux d'erreur en resubstitution.

Le pseudo-code associé à Chi2 est décrit dans la table 9.3. La formule du χ^2, utilisée lors de la décision de fusionner entre deux intervalles l et $l+1$, est :

$$\chi^2 = n_{.l} * n_{.,l+1} \sum_{k=1}^{K} \frac{(\frac{n_{kl}}{n_{.l}} - \frac{n_{k,l+1}}{n_{.,l+1}})^2}{n_{kl} + n_{k,l+1}}$$

Elle équivaut à tester si la distribution des classes dans deux intervalles adjacents est la même. Sous l'hypothèse nulle d'équivalence distributionnelle, elle suit une loi du χ^2 à $(K-1)$ degrés de liberté. Pour un risque critique α, on décide donc de fusionner si

$$\chi^2 < \chi^2_{1-\alpha}(K-1)$$

où $\chi^2_{1-\alpha}(K-1)$ représente la valeur du χ^2 ayant un degré de liberté $(K-1)$ au point de pourcentage $1-\alpha$.

```
Poser RisqueCritique = 0.5
Tant que (Inconsistence<δ)
{Répeter
  {Calculer χ² sur tous les intervalles adjacents
   Chercher Max(χ²)
   Si (Proba_Critique[Max(χ²)] ≥ RisqueCritique) Alors Fusionner
  }Jusqu'à (Proba_Critique[Max(χ²)] < RisqueCritique)
  Décrementer(RisqueCritique)
}
```

TAB. 9.3 – *Pseudo-code de l'algorithme Chi-2*

On reconnaît l'algorithme Chi-Merge [Kerber, 1992] dans la portion *Répeter...Jusqu'à* du pseudo-code de la table 9.3. En fixant une valeur limite de consistance δ, par exemple on n'accepte pas les découpages tels qu'il y ait plus de 0.05% d'erreur, ce qui permet de décrémenter itérativement le risque critique α et ainsi d'accepter plus de fusions puisque la valeur de $\chi^2_{1-\alpha}(K-1)$ augmente, les auteurs pensent ainsi dépasser la propension de Chi-Merge à produire des discrétisations trop complexes, contenant beaucoup d'intervalles à faibles effectifs. Dans la pratique, on constate la difficulté de choisir la quantité δ et la séquence de α adéquats.

9.5.3 L'optimisation est-elle vraiment pertinente ?

Nous sommes maintenant face à une alternative simple : d'un côté nous avons un algorithme très rapide donnant de bons résultats, de l'autre un algorithme un peu plus coûteux mais assurant une optimisation globale. Le passage de l'un à l'autre est-il avantageux?

Dans le cadre de l'apprentissage supervisé, un des principaux objectifs est de trouver un modèle minimisant le taux d'erreur en prédiction, celui-ci peut être estimé en appliquant le modèle proposé sur un échantillon test n'ayant pas servi à l'apprentissage. Or, on pose souvent l'hypothèse selon laquelle un modèle optimisant un critère possédant de bonnes qualités de spécialisation tout en résistant au surapprentissage sur données bruitées, se comportera mieux en prédiction. Dès lors, le problème de l'apprentissage est confiné à un problème d'optimisation [Quinlan, 1995]. Dans le cadre de cette sous-section, nous pouvons vérifier directement cette assertion en confrontant les algorithmes gloutons avec l'algorithme de Fischer.

Afin de placer les stratégies sur un pied d'égalité, nous utilisons dans tous les cas la mesure φ en fixant arbitrairement les paramètres α et λ à 0.95 et 1, afin de mesurer la pertinence des découpages. De fait, les comparaisons portent essentiellement sur les méthodes d'exploration des solutions.

	Trials	Z statistic	p-value
Fischer vs TD	300	1.54	0.1234
Fischer vs BU	300	1.26	0.2076

TAB. 9.4 – *Comparaison du nombre d'intervalles*

	Trials	Z statistic	p-value
Fischer vs TD	286	2.66	0.0077
Fischer vs BU	286	9.30	0.0000

TAB. 9.5 – *Comparaison des points optimaux choisis*

Recherche des points de discrétisation optimaux

Nous avons tout d'abord voulu caractériser les propriétés des deux algorithmes gloutons. Nous avons pour cela généré aléatoirement 300 fichiers de 50 observations d'une variable $X(.)$ avec différentes distributions et un nombre de classes d'une variable $Y(.)$ variant entre 2 et 5. Nous avons confronté les résultats obtenus (nombre d'intervalles et points de discrétisation) avec l'algorithme qui donne la solution optimale et ceux obtenus avec les deux heuristiques. Il est apparu nettement, en utilisant le test non-paramétrique de Wilcoxon de différence sur échantillon appariés avec un risque critique de 5% :

- ces deux algorithmes gloutons trouvent quasiment toujours le nombre adéquat d'intervalles $Card(U^*)$ de discrétisation quelle que soit la configuration proposée (Table 9.4);
- en revanche, tous deux échouent de manière significative lors de la détection des points de discrétisation optimaux même lorsque le nombre adéquat d'intervalles a été découvert (Table 9.5).

Gains en prédiction de l'optimalité

Certes les stratégies gloutonnes permettent de spécifier le bon nombre d'intervalles, en revanche elles échouent significativement dans la spécifications des bornes de discrétisation. Nous avons voulu quantifier les effets de cette inefficience lors de la phase de généralisation, celle qui nous intéresse le plus finalement en apprentissage automatique.

Pour ce faire, nous avons utilisé la base de données des Ondes de Breiman très largement étudiée par Lechevalier en discrétisation optimale [Réf. rapport PRC-IA]. Le programme du serveur de données UCI Irvine nous a servi pour générer 11 fichiers d'apprentissage (resp. de validation) de 300 observations (resp. 5000 observations), comportant 21 variables continues prédictives et une variable endogène à 3 classes. Nous avons construit les partitions avec les stratégies BU, qui correspond ici exactement à l'algorithme FUSINTER [Rabaseda, 1996], et

	Essai	**Moyenne**	**Ecart-type**
FISCHER	231	0.4825	0.069
FUSINTER	231	0.4807	0.070

TAB. 9.6 – *Comparaison Fusinter vs Fischer*

Fischer à partir des fichiers d'apprentissage. Notre choix en faveur de FUSINTER, plutôt qu'une stratégie "Top-Down" a été motivée par les travaux de [Zighed *et al.*, 1996] qui ont montré que sur ces fichiers de données, elle est légèrement supérieure en classement. Nous décidons alors que la classe majoritaire est la classe conclusion dans chaque intervalle, ce qui correspond à la construction d'un arbre de décision à un niveau. Nous les validons sur les fichiers tests : le taux d'erreur mesuré sert à qualifier les performances de la méthode.

Deux résultats retiennent principalement notre attention:

- à l'instar de ce que nous avions déjà mis à jour lors de nos premières expérimentations (Table 9.5), *Fusinter* trouve peu souvent la partition optimale (29 fois sur 231 essais);
- cette défection ne l'empêche pas d'être quasiment au niveau de la méthode de Fischer si l'on considère le taux d'erreur en prédiction. En effet, sur les 231 fichiers, Fusinter a été meilleur 73 fois et au même niveau 47 fois. En utilisant un test de comparaison de fréquences sur échantillons appariés, l'écart n'est pas significatif pour un risque de 5% (Tableau 9.6).

Les doutes émis par certains auteurs [Breiman *et al.*, 1984] sur la pertinence de l'optimisation en apprentissage trouvent ici un écho favorable. Dans la mesure où l'objectif est d'obtenir un faible taux d'erreur en prévision associé à un modèle le moins complexe possible en vertu du principe du Rasoir d'Occam [Russel et Norvig, 1995], il semble finalement peu intéressant de complexifier à l'extrême les méthodes d'apprentissage sachant que les gains obtenus, même s'ils sont réels (dans notre expérimentation, si nous passons à un test à 10%, on aurait conclu à une supériorité de la méthode Fischer), sont finalement minimes. Dès lors, le choix de tel ou tel procédé résulte plus de considérations pratiques comme l'interprétabilité des résultats, la simplicité du modèle ou encore la rapidité.

On peut même se demander si la notion d'optimisation en utilisant des mesures prenant en compte uniquement des tableaux de contingence sur les étiquettes est adéquate. En effet, lorsque les effectifs sont faibles, on sait que ces estimations des points de discrétisation souffrent d'une variance très élevée [Wehenkel, 1997] [de Merckt, 1993]. Dès lors, optimiser sur le fichier d'apprentissage ne sert à rien puisque l'estimateur lui-même est très instable. Nous verrons à la fin de ce chapitre quelles ont été les solutions proposées pour remédier à ce problème de variance.

En tous les cas, la conclusion à cette section est assez étonnante : en utilisant les procédures d'estimation de points de discrétisation actuelles, l'optimisation n'apporte pas de gains

significatifs en prédiction [Zighed *et al.*, 1997].

9.6 Discrétisation locale contre discrétisation globale

Par rapport aux autres méthodes symboliques, les graphes ont la propriété intéressante de pouvoir utiliser plusieurs fois sur un même chemin une discrétisation binaire simple sur la même variable. Cette discrétisation, qui rappelons-le consiste à trier les individus puis à chercher le point frontière tel que l'on optimise la mesure de qualité de partition, est nécessairement optimale sur un sommet puisque l'on teste ainsi toutes les solutions possibles. Certains auteurs maintiennent d'ailleurs [Quinlan, 1996] que cette méthode, dite locale, de discrétisation est la meilleure dans le cadre de la construction du graphe; d'autres pensent en revanche que la discrétisation en L intervalles, transformant ainsi au préalable toutes les variables continues en qualitatives ordinales, donne de meilleurs résultats [Dougherty *et al.*, 1995][48].

Dans le cadre de ce débat, nous avons recensé quatre critères :

- *la fragmentation des données* : [Wehenkel, 1997] a montré que lorsque les effectifs sont trop faibles, les estimations des points de discrétisation à l'aide du fichier d'apprentissage souffrent d'une très grande variance. Nous devons donc opérer de manière à s'éloigner le plus possible de cette situation. Un des principaux arguments des tenants de la discrétisation globale [Dougherty *et al.*, 1995] consiste justement à dire qu'une estimation des bornes sur la totalité de l'échantillon est certainement plus fiable que l'estimation sur un des sommets du graphe, contenant seulement une fraction des données de départ. [de Merckt et Quinlan, 1996] ont répliqué en montrant que les découpages binaires de variables dans les parties basses de l'arbre sont de toute manière invalidés par l'élagage qui élimine ces sommets "suspects". En revanche, ils insistent sur le fait que les découpages locaux en L intervalles posent exposent effectivement les graphes au problème de la fragmentation.

- *la complexité du graphe construit* : notre objectif est de chercher le classifieur consistant le plus simple possible. Dans le cas de la discrétisation globale, nous aurions tendance à avoir

48. Afin de donner une meilleure perspective à ce débat, il est nécessaire de signaler que [Quinlan, 1996] en fait répond explicitement à l'article de [Dougherty *et al.*, 1995]. Ces derniers avaient utilise le Gaion Ratio en discrétisation binaire, on connaît la propension de cette mesure à produire des petits découpages (le "end cut preference"). De fait, dans leur expérimentation, ils montrent que le gain ratio fragmente excessivement les données. [Quinlan, 1996] a alors répliqué en proposant une version binaire proche du principe MDLPC de [Fayyad et Irani, 1993] qu'il a validé sur un plus grand nombre de bases de données puisées sur le serveur UCI Irvine. Personnellement nous trouvons les écarts en performances minimes, cet épisode à notre sens aura surtout montré combien il était illusoire de se battre sur des centièmes de taux de succès en validation. Notons pour la petite histoire que Quinlan ira encore plus loin dans ses allégations en affirmant dans un troisième article que la discrétisation était pour lui un sujet de recherche clos [de Merckt et Quinlan, 1996].

des arbres plus en largeur mais avec une profondeur moindre. La situation est contraire avec la discrétisation locale binaire, le découpage de la même variable sur un chemin emmène souvent des arbres de plus grande profondeur.

- *la prise en compte des interactions entre les variables* : c'est le principal argument des tenants de la discrétisation locale [Quinlan, 1996]. Par exemple, l'âge moyen d'apparition de la presbytie est différent selon le sexe. Procéder à un découpage de la variable âge en fonction de la présence ou non de cette maladie hypothèque dès le départ la distinction entre les personnes de sexe opposé. Ce raisonnement trouve d'autant plus d'écho que les arbres d'induction, de par leur nature gloutonne, ont peine à détecter les interactions entre les variables [Rendell et Seshu, 1990]. Mais pour que cet avantage soit vraiment effectif, il faudrait alors adopter les stratégies de recherche en avant pour évaluer les effets conjoints de plusieurs attributs, dans la construction du graphe.

- *la rapidité de traitement* : lorsque l'on doit traiter des bases de données contenant des millions d'items, il est nécessaire de trouver des "astuces" permettant de diminuer le nombre des opérations. Or la discrétisation, rien que par la nécessité de trier les données, est très gourmande en temps de calcul. Tester tous les attributs à chaque sommet en les triant puis en essayant de trouver les meilleurs points de découpage entraîne des coûts insoutenables dans le cadre de l'extraction de connaissances dans les grandes bases de données [Catlett, 1991a]. La discrétisation globale est intéressante pour principalement deux raisons évoquées par [Liu et Setiono, 1996] : lorsqu'elle aboutit au refus de découper le domaine d'une variable, on considère que celle-ci n'est pas pertinente pour prédire la classe; le recodage conduit nécessairement à l'existence d'individus décrits identiquement, sur toutes les variables, y compris l'endogène, de fait on peut éliminer ces doublons avant de procéder à l'apprentissage proprement dit. Cette double réduction de la dimension, à la fois verticalement (sur les individus) et horizontalement (sur les variables) permet de réduire considérablement les temps de calculs.

Il est difficile de trancher entre la discrétisation globale ou locale. Si l'on raisonne en temps de calcul, la préparation des données par la discrétisation globale, s'impose. En revanche, si l'on se penche sur la précision du classifieur obtenu, les résultats obtenus par les différents auteurs sur des fichiers de benchmark de la base UCI Irvine sont contradictoires [Quinlan, 1996] [Dougherty *et al.*, 1995].

On aurait pu penser d'ailleurs que la discrétisation locale en L intervalles qui cumule deux causes menant à la fragmentation des données serait à proscrire, les tests effectués par [de Merckt et Quinlan, 1996] pour évaluer les effets de la discrétisation binaire et ternaire (les auteurs étant partis du principe qu'en médecine, on utilisait souvent la trichotomie "bas-moyen-haut" dans les variables) ont montré la supériorité de la première méthode. Dans la pratique, on

constate que la simple utilisation de mesures sensibles à la taille de l'effectif permet de produire des découpages moins complexes lorsque la taille de l'échantillon est faible [Rabaseda, 1996].

Nous nous garderons bien de conclure définitivement dans cette polémique. Chaque méthode possède ses avantages et inconvénients, les tests sur les fichiers de bases de données exemples repris chez [Quinlan, 1996] [Dougherty *et al.*, 1995] ne nous ont pas permis de montrer la supériorité de telle ou telle stratégie. Il est plus sage de penser que le choix repose surtout sur des considérations propres au domaine ou à l'expert qui jugera en fonction des critères que nous avons avancés ci-dessus.

9.7 Etude statistique de la distribution des bornes de discrétisation

Jusqu'à maintenant, nous avons considéré la recherche des points de discrétisation comme un double problème d'optimisation : optimiser une mesure de qualité de partitions sur un fichier d'apprentissage d'abord, en espérant ainsi minimiser le taux d'erreur en validation sur un fichier test. Cette deuxième étape permet ainsi d'évaluer la bonne tenue du découpage dans la population totale.

La recherche d'un point de discrétisation peut, à l'instar de tout problème où l'on essaie d'inférer sur une population à partir d'un échantillon, être ramenée à un problème de statistique mathématique simple : l'estimation. Dans cette section, nous allons nous pencher sur l'étude théorique et empirique de la distribution des points de discrétisation. Sans restreindre la portée de notre discours, nous nous cantonnerons à l'étude du découpage binaire dans un problème à deux classes qui a le mérite d'être très pédagogique.

9.7.1 Estimateur paramétrique d'un point de discrétisation

Soient $X(.)$ un attribut prédictif continu et Y la classe qui prend ses valeurs dans $\{y_1, y_2\}$. Les variables conditionnelles (X/y_1) et (X/y_2) suivent respectivement les lois statistiques $\pounds_1$ et $\pounds_2$ inconnues. Nous notons $f_{\pounds_i}$ la fonction de densité de la loi $\pounds_i$ $(i = 1, 2)$.

Proposition 12 *Si les lois $\pounds$ sont unimodales, de paramètres de localisation μ_1 et μ_2, avec $\mu_2 > \mu_1$, alors le point de discrétisation d doit minimiser*

$$\Delta = \int_d^{+\infty} f_{\pounds_1}(X/y_1) + \int_{-\infty}^{d} f_{\pounds_2}(X/y_2) \tag{9.2}$$

Le paramètre d est inconnu, nous devons l'estimer à l'aide de la statistique D.

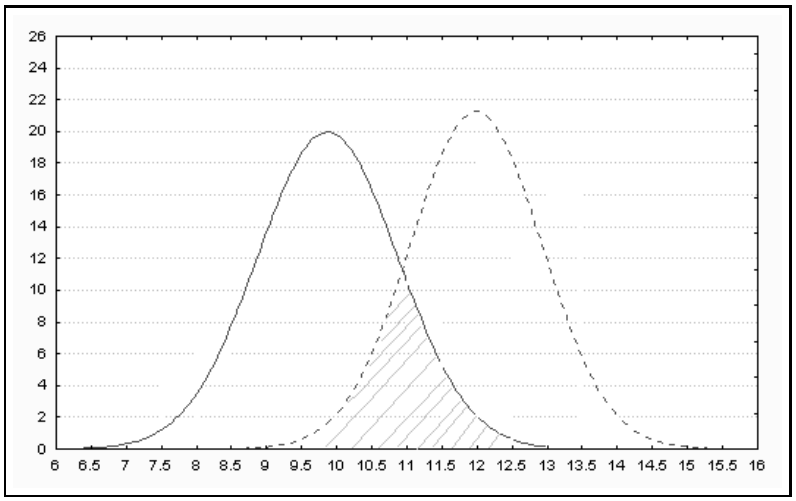

FIG. 9.8 – *X/y_1 et X/y_2 suivent deux lois normales décalées, la zone hachurée représente l'erreur à minimiser*

Etude de la loi théorique sous certaines conditions

Si nous nous plaçons dans un cadre particulier où l'on connaît les lois $\pounds_i$, nous pouvons calculer la loi théorique de D. Prenons l'exemple (graphique 9.8) où l'on a affaire à deux lois normales de paramètres d'échelle (variance) identiques :

- $X/y_1 \rightsquigarrow Normal(\mu_1, \sigma)$
- $X/y_2 \rightsquigarrow Normal(\mu_2, \sigma)$
- $\mu_2 > \mu_1$

La valeur de d voit doit minimiser la quantité Δ de l'équation 9.2. Nous allons essayer de la déterminer.

Point optimal de discrétisation dans la population

La fonction de répartition de la loi normale de paramètres (μ, σ) s'écrit :

$$\frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx$$

Sa dérivée première par rapport à x est la fonction de densité :

$$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Pour minimiser Δ, nous allons annuler sa dérivée première par rapport à d et nous assurer que la dérivée seconde est positive, indiquant que la fonction est bien convexe.

$$\begin{aligned}\frac{\partial \Delta}{\partial d} &= -\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(d-\mu_1)^2}{2\sigma^2}} + (\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(d-\mu_2)^2}{2\sigma^2}}) \\ &= \frac{1}{\sqrt{2\pi\sigma}} (e^{-\frac{(d-\mu_2)^2}{2\sigma^2}} - e^{-\frac{(d-\mu_1)^2}{2\sigma^2}})\end{aligned}$$

Donc $\frac{\partial \Delta}{\partial d} = 0$ si et seulement si

$$\begin{aligned}e^{-\frac{(d-\mu_2)^2}{2\sigma^2}} &= e^{-\frac{(d-\mu_1)^2}{2\sigma^2}} \\ -\frac{(d-\mu_2)^2}{2\sigma^2} &= -\frac{(d-\mu_1)^2}{2\sigma^2} \\ d^2 + \mu_2^2 - 2d\mu_2 &= d^2 + \mu_1^2 - 2d\mu_1 \\ d &= \frac{(\mu_1^2 - \mu_2^2)}{2(\mu_2 - \mu_1)} \\ d &= \frac{\mu_1 + \mu_2}{2}\end{aligned}$$

La dérivée seconde

$$\frac{\partial^2 \Delta}{\partial d^2} = \frac{1}{\sqrt{2\pi\sigma}} (\frac{(d-\mu_1)}{\sigma^2} e^{-\frac{(d-\mu_1)^2}{2\sigma^2}} + \frac{(\mu_2 - d)}{\sigma^2} e^{-\frac{(d-\mu_1)^2}{2\sigma^2}})$$

est toujours positive lorsque d est compris dans l'intervalle $]\mu_1, \mu_2[$.
Donc $d = \frac{\mu_1+\mu_2}{2}$ est bien la valeur minimisant l'expression Δ de l'équation 9.2.

Estimateur paramètrique de d

La quantité d ne peut pas être calculée directement puisque nous ne disposons pas généralement des valeurs μ_1 et μ_2. Nous devons utiliser un estimateur D. Celui qui vient tout naturellement à l'esprit est

$$D = \frac{\overline{X_1} + \overline{X_2}}{2}$$

où $\overline{X_i}$ sont les moyennes arithmétiques de X/y_i, estimateurs sans biais de μ_i.
D est sans biais. En effet,

$$\begin{aligned}E(D) &= E(\frac{\overline{X_1} + \overline{X_2}}{2}) \\ &= \frac{E(\overline{X_1}) + E(\overline{X_2})}{2} \\ &= \frac{\mu_1 + \mu_2}{2} \\ E(D) &= d\end{aligned}$$

La variance de D peut être calculée aisément si l'on pose une hypothèse d'indépendance des distributions conditionnelles.

$$\begin{aligned} V(D) &= \frac{1}{4}[V(\overline{X_1}) + V(\overline{X_2})] \\ &= \frac{1}{4}[\frac{\sigma^2}{n} + \frac{\sigma^2}{n}] \\ V(D) &= \frac{\sigma^2}{2n} \end{aligned}$$

9.7.2 Estimation non-paramètrique de d

Dans la pratique, tout ce que nous venions d'évoquer ci-dessus est caduque parce que nous n'avons aucune information sur les distributions conditionnelles $\pounds_i$. Nous devons donc utiliser des heuristiques qui nous rapprochent de l'optimisation de l'équation 9.2. L'optimisation du taux d'erreur en resubstitution possédant des qualités peu désirables [Breiman *et al.*, 1984], les principales méthodes connues utilisent les mesures présentées dans le chapitre sur les indicateurs d'évaluation de qualité de segmentation, notamment parce qu'elles tiennent compte de la structure de l'erreur. Le véritable problème devient alors l'étude du biais et de la variance statistique de l'estimateur, qui est le fruit d'une optimisation, ainsi construit.

Evaluation empirique des méthodes

Dans cette section, nous nous sommes inspirés des travaux de [Wehenkel, 1997] qui a utilisé une démarche proche pour évaluer la discrétisation binaire fondée sur une mesure normalisée et symétrique du gain de Shannon dont la variance asymptotique a été calculée par [Kvalseth, 1987]. Mais plutôt que de le faire sur des données réelles, l'étude de la stabilité transitoire des réseaux électriques en l'occurrence, nous avons préféré générer des données synthétiques suivant des lois statistiques que nous pouvons modifier à discrétion.

Dans cette expérimentation, nous avons testé les trois mesures suivantes en discrétisation binaire :

1. *Shannon*, parce qu'elle est la plus utilisée dans les méthodes les plus célèbres de discrétisation en apprentissage automatique [Fayyad et Irani, 1993][Quinlan, 1996];

2. *Contrast amélioré* de [de Merckt, 1993], parce qu'en combinant l'information supervisée et non supervisée, son auteur pense, même s'il n'a jamais fait d'études sérieuses là-dessus, qu'elle est de faible variabilité, notamment sur les petits effectifs;

3. Mesure d'incertitude sensible à la taille de l'échantillon de [Zighed *et al.*, 1996], parce que cette sensibilité laisse à penser qu'elle est d'une meilleure viscosité face aux fluctuations engendrées par l'échantillonnage..

Observations	[1]	[2]	[3]	D
20	9.93 ± 0.42	9.98 ± 0.42	9.94 ± 0.42	9.99 ± 0.19
50	9.94 ± 0.37	9.99 ± 0.33	9.94 ± 0.37	10.0 ± 0.12
100	9.95 ± 0.29	9.98 ± 0.27	9.95 ± 0.28	10.0 ± 0.09
500	9.99 ± 0.20	10.0 ± 0.19	9.99 ± 0.19	10.0 ± 0.03

TAB. 9.7 – *Moyenne et écart-type de l'estimation sur 200 réplications - Lois normales de mêmes variances*

Deux lois normales de même variance Dans un premier temps, nous avons repris l'exemple qui nous a servi à construire l'estimateur paramètrique. Les distributions conditionnelles sont :

- $\pounds_1 \equiv Normal(8, 1)$
- $\pounds_2 \equiv Normal(12, 1)$

La vraie valeur de d est 10. L'écart-type théorique de l'estimateur D est $V(D) = \sqrt{\frac{1}{2n}}$. Dans le tableau 9.7, nous avons noté les moyennes et écart-type du point de discrétisation mis en avant par les méthodes pré-citées pour différentes tailles d'échantillons. Le nombre de répétitions a été fixé à 200.

Les résultats montre, si cela était encore nécessaire, que les procédures paramétriques sont nettement meilleures quand il est possible de connaître les distributions conditionnelles. Pour ce qui des autres méthodes, aucune ne se démarque vraiment, elles sont convergentes, asymptotiquement sans biais, mais souffrent d'une variance trop élevée.

Deux lois normales de variances différentes Le premier exemple test était certainement trop simple pour pouvoir différencier les mesures utilisées. Dans cette deuxième série, nous générons deux lois normales décalées :

- $\pounds_1 \equiv Normal(8, 5)$
- $\pounds_2 \equiv Normal(12, 1)$

Nous ne connaissons plus les propriétés de l'estimateur D ici, il aurait fallu que l'on recommence la procédure de calcul pour développer un autre estimateur paramètrique. Ce manque de souplesse est le principal inconvénient de l'estimation paramètrique, nous l'abandonnons pour nous consacrer aux procédures plus usitées en discrétisation.. En utilisant une méthode d'optimisation numérique, la quantité d qui minimise l'équation 9.2 est égale à 10.623.

Les estimations recensées dans le tableau 9.8 sont manifestement biaisées, l'augmentation des effectifs ne corrige en rien cette propension. La variance de son côté, diminue avec la taille de

Observations	[1]	[2]	[3]
20	9.89 ± 0.77	9.61 ± 1.44	9.95 ± 0.80
50	9.88 ± 0.57	9.77 ± 0.61	10.08 ± 0.59
100	9.74 ± 0.45	9.62 ± 0.43	10.03 ± 0.43
500	9.66 ± 0.28	9.46 ± 0.30	9.95 ± 0.25

TAB. 9.8 – *Moyenne et écart-type de l'estimation sur 200 réplications - Loi normales de variance différentes*

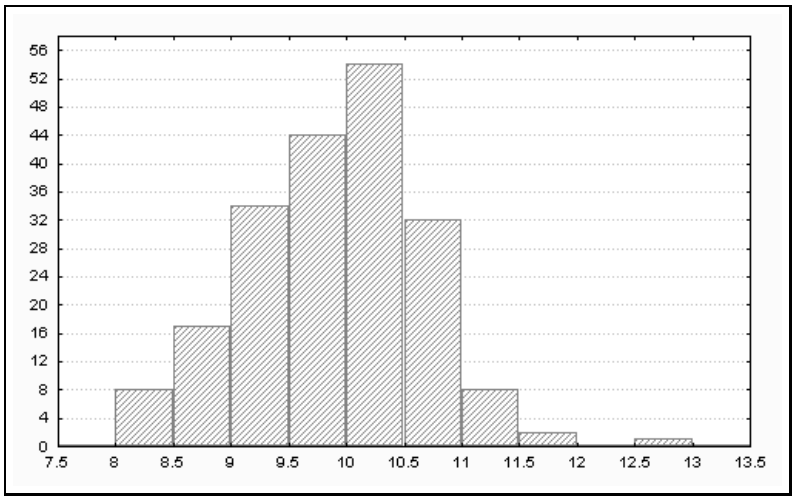

FIG. 9.9 – *Distribution empirique de l'estimateur de la borne de discrétisation par la mesure sensible à la taille de l'effectif*

l'échantillon. On retrouve les principaux résultats de simulations effectuées dans [Olszak, 1995] concernant les mesures d'association, très proches des mesures à base de gain d'entropie : dès que l'on tend, même faiblement, vers l'indépendance (il y avait 31 séquences en moyenne dans ce protocole, alors que l'on en comptait 6 dans le précédent), ces indicateurs sont extrêmement biaisés. Ce biais est alors transmis à l'estimation de la borne de discrétisation.

Ces résultats nous incitent à la prudence quant à l'utilisation de la distribution asymptotique de la mesure d'évaluation des segmentations pour conjecturer sur celle de la borne de discrétisation, notamment en ce qui concerne l'estimation de la variance [Wehenkel, 1997]. Sur deux distributions conditionnelles de paramètres d'échelle différentes par exemple, l'approximation normale est loin d'être pertinente comme le souligne le graphique où l'on représente l'histogramme de distribution de la borne estimée par la mesure sensible à la taille de l'effectif, celle qui se comporte le mieux dans notre expérimentation en terme de biais et de variance : elle est clairement non-symétrique (figure 9.9).

9.8 Conclusion

Malgré les affirmations de [de Merckt et Quinlan, 1996], le sujet est loin d'être clos. Certes, nous diposons maintenant d'une stratégie optimale de recherche mais son succès ne se traduit pas par une réduction de l'erreur en généralisation. Nous entrevoyons une raisons principale à cela : la difficiulté de définir une mesure satisfaisante qui nous rapprocherait au mieux de la borne de découpage théorique que l'on définit paramétriquement. En ce sens, les travaux de [Wehenkel, 1997] sur le biais et la variance de l'estimateur "point de discrétisation" nous semble une piste très intéressante, encore faut-il par la suite faire coïncider optimalité de la mesure et meilleure borne de discrétisation.

Chapitre 10

Agrégation de classifieurs

10.1 Introduction

La réduction de l'erreur en généralisation est une des principales motivations de la recherche en apprentissage automatique. C'est compréhensible tant le coût induit par un mauvais classement peut être élevé. En médecine par exemple, si le système expert décide, sur la base de règles extraites de l'extraction automatique de connaissances, qu'une tumeur n'est pas maligne, l'individu en question ne sera pas correctement soigné avec des conséquences qui peuvent être dramatiques. Certes, les graphes d'induction possèdent un avantage énorme sur les autre méthodes d'apprentissage : sa lisibilité, la décision peut être diagnostiquée[49] par un expert qui appréciera sa validité. Mais dans la pratique, cette procédure nous fait perdre le bénéfice de la rapidité pour le traitement à grande échelle des grosses bases de données.

Toutes les méthodes que nous avons abordées dans cette thèse visent plus ou moins à réduire cette erreur en introduisant des améliorations qui touchent à la structure du graphe ou à la pertinence des variables. Leur présumée efficacité en généralisation repose sur des intuitions qui se traduisent généralement par des effectifs plus forts sur les feuilles et une augmentation du pouvoir de représentation du classifieur afin de mieux appréhender le concept à apprendre. En réalité, si le bénéfice de ces méthodes est réel, la réduction de la taille du classifieur toutes choses égales par ailleurs entraîne une meilleure compréhension du phénomène à étudier, la relation avec la réduction de l'erreur est plutôt floue, on ne voit pas véritablement de quelle manière elles agissent. L'absence d'amélioration en généralisation sur les bases réelles, malgré toutes les réserves que l'on puisse émettre sur la représentativité de ces dernières, semblent d'ailleurs montrer que la relation est assez ténue.

[Breiman *et al.*, 1984] ont certainement été parmi les premiers à analyser formellement la dépendance entre la spécification d'un classifieur et l'erreur afférente en utilisant une analyse

49. drôle d'époque quand même où l'on diagnostique les machines plutôt que les malades

inspirée de la régression. En schématisant, l'erreur serait la somme de l'incapacité du système de représentation des connaissances à apprendre le concept sous-jacent et de la sensibilité des conclusions émises à l'échantillonnage : c'est la décomposition biais-variance. On comprend mieux ainsi les intuitions précédentes concernant les améliorations espérées des différentes stratégies promues ces vingt dernières années, le biais est réduit par l'adoption d'un système de représentation plus riche et l'augmentation de la taille moyenne des feuilles agit sur la variance. Hélas, mis à part les tentatives pour expliquer l'effet bénéfique de l'élagage [Breiman *et al.*, 1984], il n'existe pas de démonstration formelle de ces améliorations en général, mis à part pour l'agrégation des classifieurs.

Dans un rapport technique resté célèbre (suivi d'une publication deux ans plus tard [Breiman, 1996a]), [Breiman, 1994] a montré formellement en s'appuyant sur une reformulation de la décomposition biais-variance qu'il était avantageux d'adopter la décision moyenne de plusieurs classifieurs plutôt que choisir la meilleure. Il a proposé un classifieur composite construit par agrégation de prédicteurs issus de tirages boostrap [Efron et Tibshirani, 1993] dans l'échantillon d'apprentissage. Sa méthode, le bagging[50], est maintenant une référence et son idée a inspiré de très nombreux chercheurs, non seulement pour en améliorer les performances [Freund et Schapire, 1995] [Breiman, 1996b] mais également pour obtenir une meilleure estimation de l'erreur en généralisation [Efron et Tibshirani, 1995] [Tibshirani, 1996]. En fait, l'idée de l'agrégation des classifieurs, du moins en ce qui concerne les arbres de décision, n'est pas nouvelle. [Kwok et Carter, 1990], [Buntine, 1991], [Clark et Pregibon, 1992] ou encore [Oliver et Hand, 1995] avaient noté que la combinaison de plusieurs arbres donnait de meilleures résultats en généralisation, le même phénomène a été constaté par [Kononenko, 1992] pour d'autres types de classifieurs. Mais à la différence des méthodes récentes où l'on génère différents arbres en perturbant l'échantillon d'apprentissage (soit en faisant un tirage aléatoire avec tirage, soit en affectant différents poids aux individus), elles reposent sur les différentes occurrences d'un arbre construit une seule fois.

Dans ce chapitre, nous discuterons dans un premier temps du bien-fondé de l'agrégation des classifieurs en utilisant d'une part le classique schéma de décomposition biais-variance, d'autre part une proposition originale de [Heath *et al.*, 1993a]. Nous détaillerons par la suite les principales méthodes d'agrégation des classifieurs connues à ce jour. Enfin, nous discuterons des faiblesses inhérentes à ce type de méthodes, à savoir l'extrême complexité du classifieur, nous présenterons ainsi différentes manières d'y remédier. Nous conclurons alors.

50. bootstrap aggregating

10.2 Justification de l'agrégation des classifieurs

10.2.1 Décomposition biais-variance

La décomposition de l'erreur en deux termes additifs provient de l'analyse de régression, avec une variable à prédire quantitative. En classement, cette interprétation est moins évidente et on dénombre dans la littérature plusieurs interprétations [Dietterich et Kong, 1995b] [Breiman, 1996b] [Tibshirani, 1996] [Kohavi et Wolpert, 1996]. Il reste qu'au final l'idée sous-jacente est toujours la même : le biais traduit l'incapacité du modèle à apprendre correctement le concept i.e l'erreur systématique commise; la variance, la sensibilité de l'algorithme à l'échantillon d'apprentissage, on dit qu'une méthode souffre d'une grande variance si une faible perturbation dans l'échantillon implique d'importantes modifications du classifieur.

Dans ce qui suit, nous présenterons tout d'abord les calculs pour la régression, ils sont relativement simples et aisés à appréhender. Nous montrerons ensuite une traduction possible pour des fonctions d'erreur de type "0-1".

Décomposition pour une fonction d'erreur quadratique

Dans le schéma de régression classique, on cherche à reconstruire un concept $f(x)$ tel que $y = f(x) + \varepsilon$, où x représente un vecteur de variables exogènes, y la variable endogène et ε un bruit quelconque. La fonction $\widehat{f}(x)$ est estimée sur un échantillon d'apprentissage Ω^a. Pour juger de son efficience, on utilise une fonction d'erreur quadratique qui s'écrit

$$Erreur(f) = E_{\Omega^a}[\widehat{f}(x) - f(x)]^2 \tag{10.1}$$

Soient maintenant $\Omega^a_{(1)}, \ldots, \Omega^a_{(s)}$, s échantillon extraits de manière indépendante dans la population Ω, produisant chacun un prédicteur $\widehat{f}_i(x)$ $(i = 1, \ldots, s)$. Le prédicteur agrégé $\overline{\widehat{f}}(x)$ est défini par

$$\overline{\widehat{f}}(x) = \lim_{s \to \infty} \frac{1}{s} \sum \widehat{f}_i(x)$$

dont l'erreur quadratique moyenne s'écrit

$$Biais(f) = E_{\Omega^a}[\overline{\widehat{f}}(x) - f(x)]^2 \tag{10.2}$$

[Breiman, 1996b] montre que

$$Erreur(f) = E_{\Omega^a}(\varepsilon^2) + Biais(f) + Var_{\Omega^a}(f) \tag{10.3}$$

où $Var_{\Omega^a}(f) = E_{\Omega^a}[\widehat{f}(x) - \overline{\widehat{f}}(x)]^2$.

C'est la décomposition fondamentale de l'erreur en biais et variance pour le schéma de régression avec une erreur quadratique. On remarque que tous les termes de l'équation 10.3 étant positifs, l'erreur moyenne du prédicteur agrégé (équation 10.2) sera toujours inférieure à l'erreur moyenne d'un seul prédicteur 10.1.

Décomposition pour une fonction d'erreur de type "0-1"

La transposition pour une fonction d'erreur de type "0-1" (0 si pas d'erreur, 1 sinon) est le champ d'un large débat, [Friedman, 1996] recense plusieurs approches, le vrai problème est de construire une décomposition qui se rapproche le plus de l'équation 10.3. Parmi les méthodes proposées, nous choisirons la formulation de [Dietterich et Kong, 1995a] parce qu'elle possède la particularité d'être facilement calculable. En revanche, elle a le défaut de présenter une variance négative dans certains cas, ce qui est rédhibitoire pour certains auteurs [Kohavi et Wolpert, 1996] [Gavin, 1997]. Nous discuterons plus en détail de cette dernière remarque plus loin.

Maintenant la variable endogène $Y(.)$ est qualitative, elle prend ses valeurs dans $\{y_1, \ldots, y_K\}$. La fonction d'erreur du classifieur $\widehat{f}$ s'écrit pour un individu ω à classer

$$e(\omega) = \begin{cases} 0 \ si \ \widehat{f}(\omega) = f(\omega) \\ 1 \ si \ \widehat{f}(\omega) \neq f(\omega) \end{cases}$$

Si nous disposons de s échantillons d'apprentissage indépendants, l'erreur moyenne du prédicteur agrégé pour un individu ω est tout simplement la fréquence de mauvais classement sur les s classifieurs $\widehat{f}_i()$

$$e_A(\omega) = \lim_{s \to \infty} \frac{1}{s} \sum_{i=1}^{s} e_i(\omega)$$

Cette quantité constitue également l'estimateur de la probabilité d'occurrence d'un mauvais classements sur un individu lorsque l'on utilise un prédicteur $\widehat{f}(\omega)$. En effet, il est licite de penser que les prédicteurs étant construits avec des échantillons indépendants, avec des préférences identiques déterminées a priori, ils suivent la même loi de distribution, en particulier pour le taux d'erreur en généralisation associé.

De fait, nous pouvons écrire l'erreur moyenne d'un algorithme d'apprentissage produisant l'estimateur $\widehat{f}()$

$$E[\widehat{f}(\omega) \neq f(\omega)] = e_A(\omega)$$

Nous devons alors décomposer cette erreur en deux portions, le biais et la variance. Soit une observation ω à classer, si $e_A(\omega) > \frac{1}{2}$, le prédicteur agrégé le classera mal, [Dietterich et Kong, 1995b] relie le biais à la notion d'erreur systématique i.e l'incapacité du classifieur en moyenne de bien classer un individu[51]. Dès lors, le biais s'écrit

$$Biais[\widehat{f}, \omega] = \begin{cases} 0 \ \ si \ \ e_A(\omega) \leq \frac{1}{2} \\ 1 \ \ si \ \ e_A(\omega) > \frac{1}{2} \end{cases}$$

51. Notons que dans [Dietterich et Kong, 1995a], les mêmes auteurs adoptent une expression plus restrictive à l'erreur systématique et utilise la condition $e_A(\omega) > 0$. Dans la pratique, il s'est avéré que ce type de spécification était trop restrictif en faisant l'amalgame entre prédicteur idéal (zéro erreur) et le prédicteur bayésien (erreur limite).

et la variance s'obtient par différence

$$Var[\widehat{f},\omega] = E[\widehat{f}(\omega) \neq f(\omega)] - Biais[\widehat{f},\omega]$$

Le plus grand reproche que l'on peut adresser à cette formulation est la possibilité d'occurrence de variance négative sur une observation donnée. [Dietterich et Kong, 1995a] le justifie par le fait qu'un point souvent mal classé par un algorithme peut, dans certains cas, être bien classé par quelques classifieurs $\widehat{f}_i()$. Ainsi, ces occasionnels classements chanceux réduisent l'erreur moyenne.

Quoiqu'il en soit, nous pensons que cette formulation est extrêmement séduisante car elle permet un calcul direct dans les expérimentations et de quantifier ainsi les effets d'une modification sur le biais et la variance. C'est ainsi que [Dietterich et Kong, 1995b] ont constaté empiriquement que :

- les techniques d'élagage jouent très peu sur la variance tout en accroissant le biais,
- les techniques d'agrégation augmentent très légèrement le biais mais cela est compensé par une réduction drastique de la variance.

10.2.2 Réduction de l'erreur théorique

La décomposition de biais-variance de l'erreur est un paradigme fort qui a séduit de nombreux chercheurs pour expliquer le rôle bénéfique de l'agrégation des classifieurs. Mais elle n'est pas la seule manière de prouver qu'un prédicteur agrégé est meilleur en généralisation, [Heath *et al.*, 1993a] ont produit une démonstration formelle de la réduction de la probabilité de mal classer sous certaines conditions mais en s'appuyant cette fois-ci sur l'analyse de la combinaison de l'erreur sous l'hypothèse d'indépendance entre les classifieurs.

Soit $b(\widehat{f})$ la probabilité de bien classer d'un classifieur $\widehat{f}$ construit sur un échantillon quelconque tiré de la population Ω, nous le noterons b afin de simplifier la lecture. Nous nous posons la question de savoir quelle serait la probabilité de mal classer de s classifieurs $\widehat{f}_i$ où la décision serait prise à la majorité. Sans restreindre la porté de notre analyse, nous nous plaçons dans le cadre d'un problème à deux classes et d'un nombre de classifieurs pair, de fait le vote à la majorité revient à choisir la classe y_k telle que

$$\sum_{i=1}^{s} [\widehat{f}_i(\omega) = y_k] > \frac{s}{2}$$

où $[\widehat{f}_i(\omega) = y_k]$ est une fonction indicatrice

$$[\widehat{f}_i(\omega) = y_k] = \begin{cases} 1 \; si \; \widehat{f}_i(\omega) \neq y_k \\ 0 \; si \; \widehat{f}_i(\omega) = y_k \end{cases} \tag{10.4}$$

Or nous savons qu'un classifieur $\widehat{f}_i$ quelconque peut bien classer selon une probabilité ε. La variable indicatrice (équation 10.4) suit une loi de probabilité de Bernouilli de paramètre $[\widehat{f}_i(\omega) = y_k] \rightsquigarrow Binomial(1, b)$. Les classifieurs étant par hypothèse indépendants et identiquement distribués (construits sur des fichiers issus aléatoirement de la même population, même méthode choisie arbitrairement), la variable $\widehat{f}_A = \sum_{i=1}^{s} [\widehat{f}_i(\omega) = y_k]$ suit une loi binomiale $Binomial(s, b)$.

De fait, la probabilité de bien classer du prédicteur agrégé s'écrit :

$$P(\widehat{f}_A > \frac{s}{2}) = 1 - \sum_{j=0}^{s/2} C_s^j \, b^j \, (1-b)^{s-j} \tag{10.5}$$

Dans l'équation 10.5, j représente le nombre de prédicteurs classant bien l'individu ω. L'agrégation n'est bénéfique que si

$$P(\widehat{f}_A > \frac{s}{2}) - b > 0$$

i.e l'introduction de plusieurs classifieurs augmente le taux de reconnaissance.

Un simple graphique pour $s = 10$, nous permet de voir l'évolution de cette différence selon les valeurs de b (figure 10.1); on constate que l'amélioration joue dans les deux sens :

- lorsque le taux de bon classement individuel est légèrement supérieur à approximativement 0.5, l'effet induit est très fort pour décélérer quand on s'approche de la valeur $b = 1$;
- en revanche, quand les classifieurs individuels sont mauvais, leur agrégation entraîne des performances encore pires.

Il apparaît au regard de cette analyse que l'effet de l'agrégation est multiplicatif, il améliore les bons classifieurs et détériore les mauvais, de plus lorsque nous nous approchons de la limite théorique de bon classement, il n'amène pas grand chose, ce qui semble assez intuitif.

10.2.3 Formule des probabilités totales

On sait que le prédicteur bayésien est optimal au sens où il est celui qui minimise l'erreur pour une matrice de coûts symétrique unitaire, pour un individu ω de la population Ω à classer, on lui affectera la classe qui maximise sa probabilité a posteriori, c'est celui que l'on essaie d'approximer en général

$$\widehat{y} = \arg\max P(y_k/\omega) \tag{10.6}$$

Lorsque l'on veut classer un individu avec un classifieur M_i, on choisit en général la classe qui maximise l'expression

$$\widehat{y} = \arg\max_k P(y_k/\omega, M_i) \tag{10.7}$$

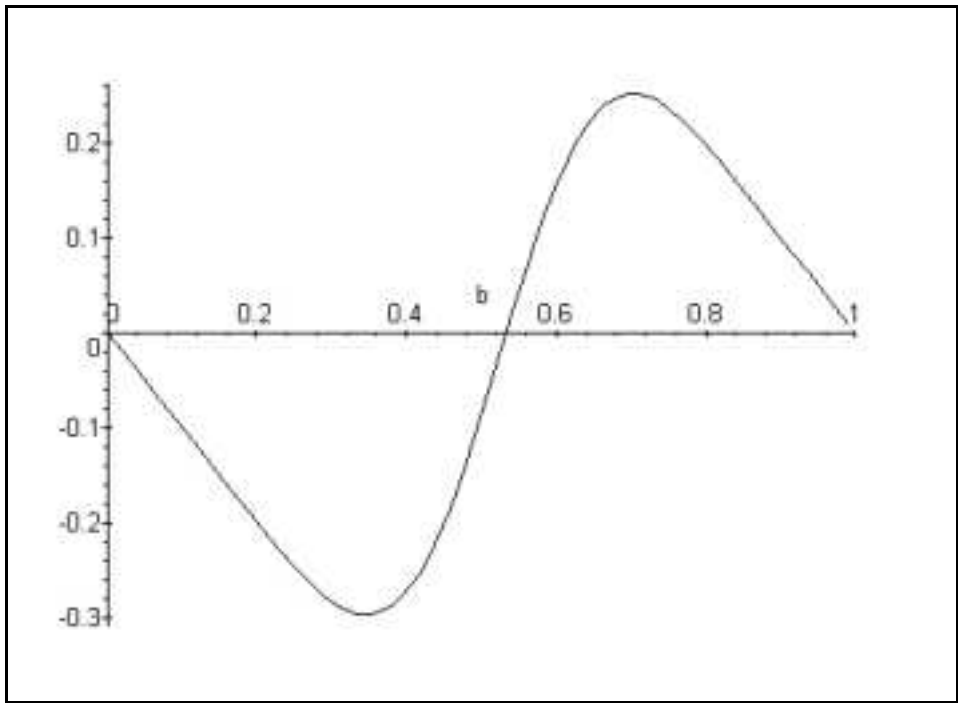

FIG. 10.1 – *Effet multiplicatif de l'agrégation de 20 classifieurs*

i.e on lui affecte la classe qui est la plus probable sachant le sous-groupe (la feuille) à laquelle il appartient, notre objectif bien entendu est d'approximer au mieux le prédicteur. La relation entre les deux probabilités des équations (10.6) et (10.7) nous est fournie par le théorème des probabilités totales

$$P(y_k/\omega) = \sum_i P(M_i/\omega) \times P(y_k/\omega, M_i) \tag{10.8}$$

Puisqu'il est impossible de générer tous les arbres, toute la problématique de l'agrégation des classifieurs repose sur le choix d'un ensemble de classifieurs les plus intéressants M_i sur lesquels nous estimons la quantité $P(M_i/\omega)$. Dans le cas du bagging que nous détaillerons plus loin par exemple, tous les modèles sont supposés équiprobables puisque créés avec les mêmes préférence sur des échantillons de caractéristiques similaires.

10.3 Agrégation par apprentissage sur un seul fichier

Les méthodes décrites dans ce chapitre proposent le même point de vue : définir un ensemble d'arbres M_i sur lesquels on calculera les quantités $P(M_i/\omega)$ à partir d'un seul échantillon Ω^a. Elles diffèrent donc uniquement de la manière d'extraire des cas particuliers d'arbres.

10.3.1 Moyennage

Le moyennage[52] représente, avec les arbres à options, un méthode à part dans toutes les stratégies d'agrégation de classifieurs. On ne génère qu'un seul arbre à partir des données, mais en révisant les probabilités d'affectation lorsque l'on a un individu à classer. Pour ce faire, on définit des séries de sous-arbres sur lesquels on suppute différentes qualités (biais, variance) à partir de l'arbre de référence, celui qui a été construit entièrement et optimisé sur l'échantillon d'apprentissage.

Afin de mieux illustrer notre propos, nous présentons deux méthodes de construction de l'ensemble de sous-arbres. Nous remarquerons qu'elles sont l'apanage des tenants de l'approche bayésienne, ou du moins qui en sont proches.

L'ensemble de chemins

[Buntine, 1992] définit l'ensemble de chemins[53] comme l'ensemble des arbres construits de la manière suivante :

1. soit $PSet$ l'ensemble des arbres à construire, au départ il ne contient que l'arbre initial optimisé;
2. pour un individu ω à classer, tracer le chemin le long de l'arbre jusqu'à la feuille;
3. élaguer chaque noeud pré-terminal itérativement le long de ce chemin, tous les arbres ainsi constitués sont introduits dans $PSet$ (cf. exemple figure 10.2).

$P(M_i/\omega)$ et $P(y_k/\omega, M_i)$ sont ensuite calculées à partir des arbres appartenant à $PSet$. L'idée sous-jacente, bien qu'elle n'ait jamais été formulé ainsi par l'auteur, est qu'à mesure que l'on remonte, la variance de l'erreur sur les sommets terminaux diminue.

L'ensemble éventail

[Oliver et Hand, 1995] prennent le contre-pied de la méthode précédente en élaborant une méthode où l'on essaie plutôt de trouver des séquences d'arbres où le biais est plus faible même si c'est au détriment de la variance : ce sont les ensembles d'arbres en éventail[54].

La démarche est la suivante :

1. soit $Fset$ l'ensemble initial ne contenant qu'un singleton, l'arbre optimisé, donc après l'élagage de la méthode;

52. averaging
53. path set
54. fanned set

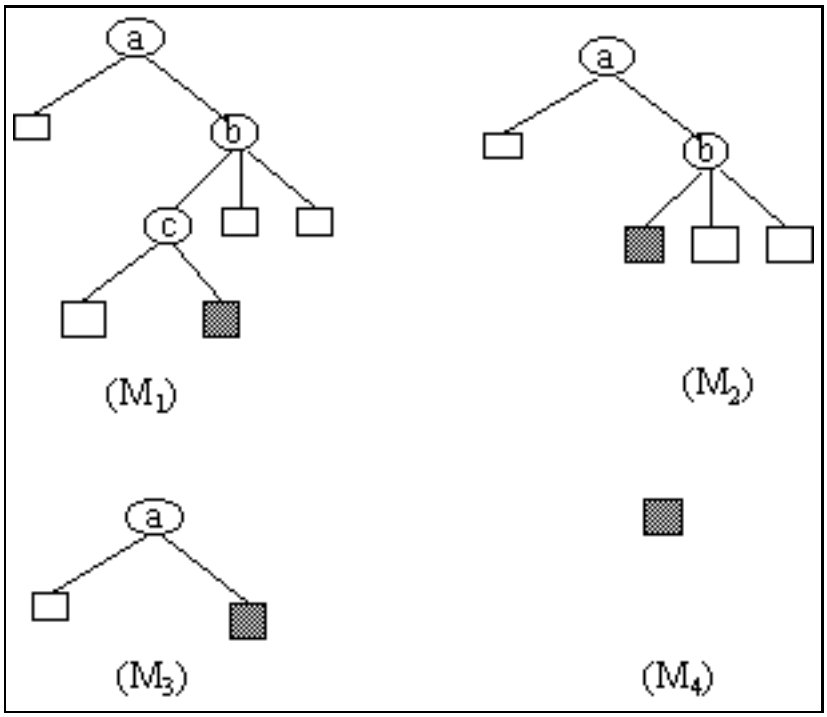

FIG. 10.2 – *Les feuilles en grisé sont celles où se situe l'individu. En remontant vers la racine, on définit une série d'arbres,* $PSet = [M_1, M_2, M_23, M_4]$

2. pour un individu ω à classer, à partir de la feuille qui le contient, on essaie toutes les segmentations à un niveau, les arbres résultants sont introduits dans $Fset$ (cf. exemple figure 10.3).

Les mêmes auteurs ont par la suite introduit une amélioration à leur algorithme [Oliver et Hand, 1995], en générant tous les $FSet$ correspondant à chaque élément de $PathSet$.

Calcul des probabilités

Les probabilités $P(y_k/\omega, M_i)$ sont calculées facilement à l'aide des fréquences d'occurrence des classes sur un noeud terminal, on peut adopter une approche plus raffinée en utilisant des estimations laplaciennes [Buntine, 1991].

La détermination de $P(M_i/\omega)$ est en revanche un peu plus compliquée, il est clair que la solution passe par une formulation bayésienne qui offre un cadre théorique cohérent pour y répondre

$$P(M_i/\omega) = \frac{P(M_i) \times P(\omega/M_i)}{P(\omega)} \tag{10.9}$$

Puisque $P(\omega)$ est constant, on peut normaliser l'équation 10.9 par la somme sur l'ensemble

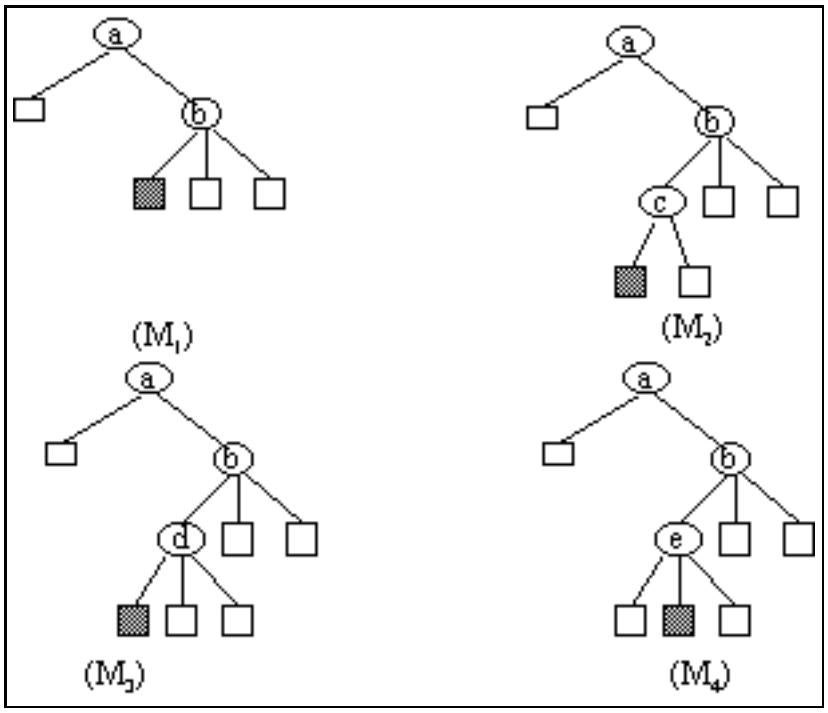

FIG. 10.3 – *Les feuilles en grisé sont celles où se situe l'individu ω. A partir de l'arbre M_1, on définit une série d'arbres consituée par les segmentations alternatives, $FSet = [M_1, M_2, M_23, M_4]$*

des modèles extraits de la quantité au numérateur

$$P(M_i/\omega) = \frac{P(M_i) \times P(\omega/M_i)}{\sum_i P(M_i) \times P(\omega/M_i)} \tag{10.10}$$

Le calcul de la probabilité a priori d'apparition du modèle est maintenant bien maîtrisé, il existe dans la littérature pléthore de distributions qui sont plus ou moins adaptées aux caractéristiques du domaine d'étude [Wehenkel, 1992] [Wallace et Patrick, 1993]. Une des plus simples, indiquant une préférence à la simplicité, est la distribution de type II de [Buntine, 1992] (avec $w > 0$)

$$P(M_i) \propto w^{(Nombre\_Feuilles + Nombre\_Noeuds)}$$

Enfin, pour estimer la vraisemblance $P(\omega/M_i)$, [Oliver et Hand, 1995] proposent d'utiliser le produit d'une fonction Beta de dimension K pour estimer l'occurrence sur les L feuilles que comporte l'arbre

$$P(\omega/M_i) = \prod_{l=1}^{L} \frac{\prod_{k=1}^{K} n_{kl}!}{(\sum_{k=1}^{K} n_{kl})!}$$

[Oliver et Hand, 1995] ont comparé favorablement leur méthode à la référence C4.5 de [Quinlan, 1993a], leur principal avantage est que l'on peut une fois pour toutes recalculer toutes les distributions de probabilités des classes sur les feuilles avant la généralisation. [Buntine, 1991] argue également

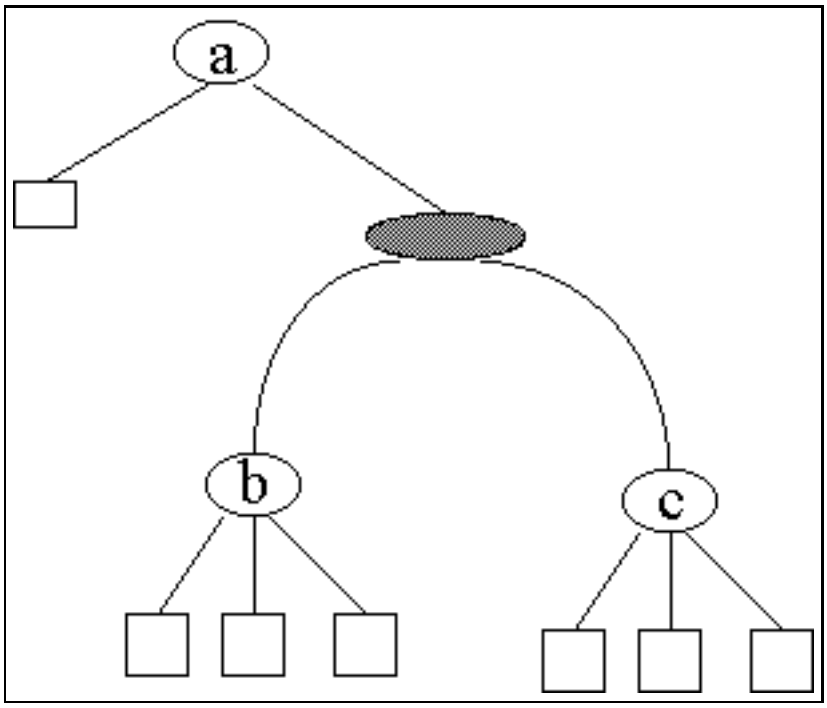

FIG. 10.4 – *Segmentations alternatives sur le sommet grisé: l'une avec l'attribut b, la seconde avec l'attribut c*

que ces probabilités sont nettement plus réalistes que les estimations naïves (fréquences, estimateur laplacien...) et peuvent être utilisées telles quelles pour l'évaluation des coûts de décision et des risques.

10.3.2 Arbres à options

Toujours dans l'optique de générer un seul arbre pour la construction d'un classifieur interprétable, [Buntine, 1991] relayé par [Kohavi et Kunz, 1997] a proposé les arbres à options[55].

Ces auteurs partent du constat que le choix de l'attribut de découpage sur un noeud se décide souvent à très peu de choses, un ou deux individus supplémentaires peuvent faire basculer la décision, [Kohavi et Kunz, 1997] notamment pensent que les structures de séries d'arbres produits par rééchantillonage que nous verrons dans la section suivante diffèrent essentiellement sur ces noeuds. Dès lors, plutôt que de décider en faveur de l'attribut qui maximise la mesure d'évaluation des segmentations, ils proposent de restituer l'incertitude en proposant plusieurs découpages alternatifs sur un noeud. L'arbre par la suite maintiendra ces branches en parallèles, il est possible que d'autres options voient le jour plus bas.

L'objectif avoué ici est bien de réduire la variance en l'absorbant dans la structure de l'arbre. En choisissant le nombre adéquat d'options sur un noeud, les variations de l'effectif ont moins de

55. option trees

prise sur la décision finale puisque les solutions alternatives sont maintenues le long de l'arbre. De plus on améliore l'exploration de l'espace des solutions, on peut considérer les arbres à options comme une variante de la recherche en avant limitée avec plusieurs solutions en parallèle[56] [Ragavan *et al.*, 1993b].

Le succès de la méthode repose sur les deux éléments essentiels que sont le choix du groupe de variables introduites comme options sur un noeud, et la combinaison des solutions extraites sur les feuilles.

Sur la première question, il n'y a pas de règle véritable dans la littérature. [Kohavi et Kunz, 1997] utilisent une règle empirique qui consiste à multiplier le meilleur gain ($G_{\max}$) par un facteur a, et à ne choisir que les variables qui possèdent un gain inférieur à $a \times G_{\max}$. En faisant varier la valeur de a, on acceptera plus ou moins les solutions alternatives. Dans le même ordre d'idées, il semble naturel d'utiliser un test de significativité de la segmentation et de ne garder que les attributs qui rejettent l'hypothèse nulle d'indépendance avec la variable à prédire.

Pour ce qui est de la deuxième question, on a le choix entre un vote pondéré et un vote à la majorité simple. [Buntine, 1991] toujours fidèle à la tradition bayésienne propose des méthodes de calculs analogues au moyennage d'arbres; [Kohavi et Kunz, 1997], plus pragmatique, préfère le vote à la majorité simple et montre que sa méthode se comporte au moins aussi bien que le bagging. Il reste un écueil de taille cependant, ces performances se font souvent au ironiquement détriment de la lisibilité, le cheval de bataille de l'auteur, car l'arbre induit est de trop grande taille.

10.3.3 Construction aléatoire

[Kwok et Carter, 1990] sont les principaux promoteurs de cette stratégie qui historiquement est certainement la première qui ait abordé explicitement l'agrégation des classifieurs pour les arbres de décision. Partant toujours du constat que les segmentations concurrentes sont très proches sur un noeud, ils proposent de choisir au hasard parmi les q meilleures segmentations : c'est le processus de construction aléatoire. On peut ainsi construire un grand nombre en utilisant le même fichier d'apprentissage Ω^a. L'agrégation se fait par la suite par un vote à la majorité simple, l'hypothèse est donc l'équiprobabilité des arbres, on expecte une qualité égale des arbres produits (figure 10.5). [Dietterich et Kong, 1995b] ont favorablement comparé cette méthode avec le bagging classique, son action est toujours de réduire la variance.

A l'instar de ce qui a été dit pour les arbres à options, l'efficacité de la randomisation dépend du choix du nombre q, qui elle-même dépend de plusieurs facteurs. Mais, on peut tout aussi bien mettre en oeuvre les tests statistiques de significativité lors de la sélection des attributs.

56. beam search

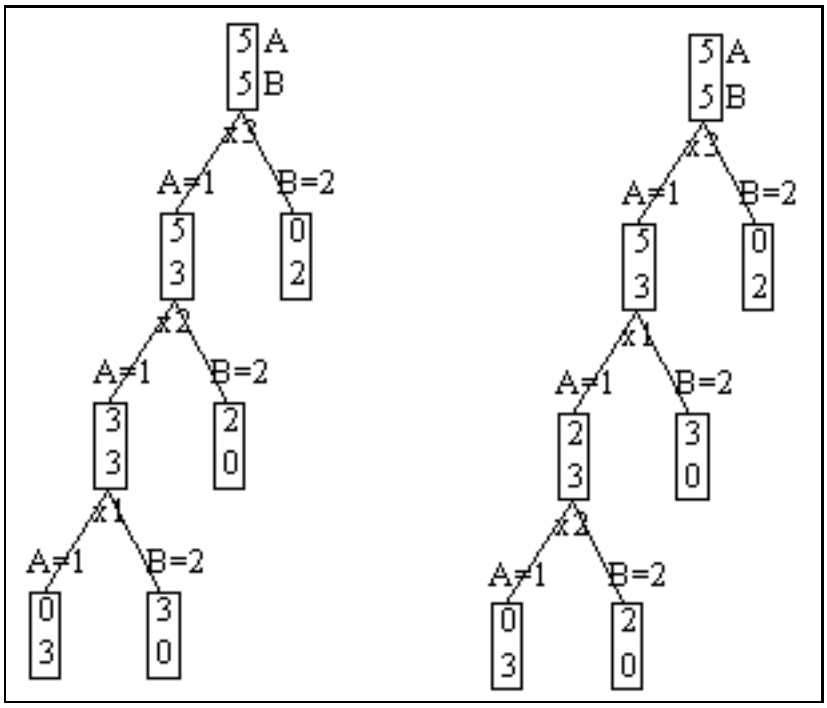

FIG. 10.5 – *Le choix alternatif sur le sommet à gauche au premier niveau montre que la qualité de l'arbre final n'en est pas affectée*

10.4 Agrégation par apprentissage sur plusieurs fichiers différents

A la différence de la section précédente, le classifieur agrégé est construit à l'aide d'apprentissages répétés sur des échantillons $\Omega^a_{(1)}, \ldots, \Omega^a_{(s)}$. Les méthodes diffèrent ici par la constitution des $\Omega^a_{(i)}$ $(i = 1, \ldots, s)$ à partir de l'échantillon initial Ω^a et par le choix de la distribution $P(M_i/\omega)$ qui en résulte.

10.4.1 Bagging

Cette méthode de [Breiman, 1996a] est la référence comme peut l'être le tirage aléatoire simple en statistique. Pour les s classifieurs à produire, on effectue un tirage aléatoire avec remise dans Ω^a, puis l'on construit le classifieur sur l'échantilon $\Omega^a_{(i)}$ ainsi constitué. On remarquera que $|\Omega^a_{(i)}| = |\Omega^a| = n$, mais qu'environ 37% des individus de Ω^a sont absents de $\Omega^a_{(i)}$. En effet, au premier tirage, un individu a $(1 - \frac{1}{n})$ probabilité de ne pas être choisi. Au bout de n tirages indépendants et équiprobables, la probabilité qu'il n'ait jamais été choisi est

$$\begin{aligned}(1 - \frac{1}{n})^n &\approx e^{-1} \\ &\approx 0.368\end{aligned}$$

Enfin, le schéma de tirage étant simple et avec remise, on conjecture que les M_i sont équi-

probables au regard de l'individu ω à classer i.e

$$P(M_i/\omega) = \frac{1}{s} \tag{10.11}$$

En résumé la procédure de construction du prédicteur agrégé est la suivante :

1. pour s replications
   (a) construire un échantillon $\Omega^a_{(i)}$ de taille n en effectuant un tirage aléatoire avec remise dans Ω^a
   (b) construire le classifieur M_i à partir de $\Omega^a_{(i)}$
2. on affecte alors en généralisation à l'individu ω la classe y_{k^*} tel que

$$y_{k^*} = \arg\max_k \sum_i \frac{1}{s} [M_i(\omega) = y_k]$$

ce qui correspond à un vote à la majorité simple[57].

L'efficacité du bagging sur des fichiers benchmark n'est plus à démontrer. [Dietterich et Kong, 1995b], [Breiman, 1996a] et [Quinlan, 1996] ont tous constaté empiriquement qu'il améliorait dans de grandes proportions les performances des méthodes classiques de construction de graphes; de plus, sa convergence est rapide, au delà de 25 itérations les gains sont marginaux. Pour notre part, nous la trouvons extrêmement séduisante parce qu'elle repose sur une cohérence théorique solide, notamment dans le choix de la distribution 10.11.

10.4.2 Boosting

Mais la "beauté" théorique est une chose, la performance en est une autre. En statistique, on s'est rendu compte depuis longtemps que des schémas de tirages plus élaborés (par quota, par strates...) se révélaient plus précis dans la pratique, dans le même ordre d'idée certains chercheurs se sont demandés s'il était possible de mieux choisir les classifieurs dans l'espace des hypothèses afin "d'éparpiller l'erreur" [Gavin, 1997].

[Freund et Schapire, 1995] ont proposé un système de pondération, le boosting, qui permettrait de construire une série de prédicteurs couvrant mieux, par rapport au bagging, l'espace de représentation [Dietterich, 1997]. La procédure est la suivante :

1. au départ, on affecte à chaque individu le poids $w_\omega = \frac{1}{n}$ pour le premier échantillon $\Omega^a_{(1)} = \Omega^a \ (i = 1)$;

57. la quantité $[M_i(\omega) = y_k]$ vaut 1 si $M_i(\omega) = y_k$, 0 sinon.

2. on construit le premier classifieur M_i en introduisant explicitement les poids w_ω [58], ce qui permet d'isoler un ensemble d'individus mal classés en apprentissage $\Omega^{a,mc}_{(i)} \subset \Omega^a$, avec $\varepsilon_i = \frac{card(\Omega^{a,mc}_{(i)})}{card(\Omega^a)}$;

3. $\varepsilon_i \geq 0.5$ ou $\varepsilon_i = 0$ sont des conditions d'arrêt de l'algorithme;

4. dans le cas contraire, on pondère chaque individu bien classé par le modèle M_i de l'échantillon Ω^a par la quantité $\beta_i = \frac{\varepsilon_i}{1-\varepsilon_i}$ pour constituer l'échantillon $\Omega^a_{(i+1)}$ (les autres individus possèdent le poids $\frac{1}{n}$). Le tout est renormalisé de manière à obtenir $\sum_{\omega=1}^{n} w_\omega = 1$, on remarque ici que les poids w_ω ne sont donc plus uniformes sur l'ensemble d'apprentissage;

5. on recommence à l'étape 2 ($i = i + 1$);

6. on agrégera alors les s classifieurs M_i en les pondérant par la quantité $\log(\frac{1}{\beta_i})$, on affectera ainsi en généralisation la classe y_{k^x} à l'individu ω si et seulement si

$$y_{k^x} = \arg\max_k \sum_i \log(\frac{1}{\beta_i}) \, [M_i(\omega) = y_k]$$

L'estimation du poids du classifieur est le principal écueil de la méthode, en effet ici on pose

$$P(M_i/\omega) = \log(\frac{1}{\beta_i}) \tag{10.12}$$

Intuitivement, on comprend que l'on pénalise plus les mauvais classifieurs dans le vote : plus ε_i est élevée, plus petite sera la quantité $\frac{1}{\beta_i}$. En revanche, du point de vue de l'analyse probabiliste, l'équation 10.12 est indéfendable, elle ne répond même pas à la propriété fondamentale

$$\sum_i P(M_i/\omega) = 1$$

Néanmoins, malgré ces réserves, il reste que les performances du boosting sont meilleures que celles du bagging en moyenne, son défaut étant une grande variabilité selon les fichiers d'apprentissage. De nombreux chercheurs travaillent soit dans le sens de la réduction de cette instabilité [Quinlan, 1996], soit dans le sens de la recherche d'une meilleure heuristique de pondération (des individus et/ou des classifieurs) afin d'améliorer les performances [Breiman, 1996b] [Schapire et Freund, 1996] [Gavin *et al.*, 1997]. Dans le même ordre d'idée, on peut également se poser la question de savoir si d'autres stratégies d'agrégation des préférences ne seraient pas plus appropriées lors de la prise de décision, on pense notamment aux méthodes multicritères [Auray *et al.*, 1993].

Notons que pour que ces méthodes agissent convenablement, il faut que la source de l'erreur provienne essentiellement de la variance. Dans le cas contraire, lorsque le concept est déterministe

58. on peut se réferer, entre autres, aux travaux de [Breiman *et al.*, 1984] pour l'intégration des poids et des coûts dans l'induction de graphes

ou que le classifieur est extrêmement stable, on ne peut espérer de réduction de l'erreur par agrégation de classifieurs. Dans certains cas même, l'augmentation concomitante du biais entraîne son accroissement. Sous cet angle, on comprend mieux les meilleures performances en moyenne du boosting face au bagging, dans ce dernier on ne compte que sur la variabilité du modèle, alors que dans le boosting on introduit de la variance supplémentaire en modifiant les poids relatifs des individus. Cela explique que parfois, lorsque les contraintes de poids sont abusives, on introduit des prédicteurs de mauvaise qualité qui détériorent les performances de l'ensemble.

10.5 Réduction des classifieurs agrégés

Toutes les heuristiques de la section 10.4 ainsi que certaines de la section 10.3, même si elles donnent de bons résultats en généralisation souffrent d'un défaut que nous qualifierons de rédhibitoire : celui de produire un classifieur illisible. Les règles sont enchevêtrées et se recouvrent, il paraît difficile de distinguer véritablement les causalités en jeu pour comprendre l'affectation d'une classe à un individu. Nous perdons ici tout le bénéfice des graphes d'induction qui, en plus du classement, proposent un modèle explicatif. De fait, on peut se demander si les gains obtenus en performances sont véritablement intéressants dans ce cas, et tant qu'à obtenir des prédicteurs boîtes noires pourquoi ne pas utiliser des familles de classifieurs moins explicites mais plus riches comme les réseaux de neurones. Certes, il existe dans la littérature plusieurs tentatives pour réduire le nombre de prédicteurs des classifieurs agrégés, mais l'optique est plutôt l'optimisation des performances et non la lisibilité du classifieur [Margineantu et Dietterich, 1997].

Dans cette section, nous détaillerons deux voies différentes pour revenir à la simplicité de structure de la fonction de classement issue des graphes d'induction tout en bénéficiant des avantages de l'agrégation des classifieurs. La première est originale, elle consiste tout simplement à utiliser des procédures de simplification sur la base de règles agrégée; la seconde consiste à utiliser le prédicteur agrégé pour constituer un échantillon de taille virtuellement infinie et profiter ainsi d'une propriété méconnue : sous certaines restrictions, lorsque la taille de l'échantillon d'apprentissage tend vers l'infini, le taux d'erreur en généralisation des graphes converge vers l'erreur du prédicteur bayésien [Breiman *et al.*, 1984].

10.5.1 Réduction par simplification des bases de règles

Dans cette section, nous avons voulu tester l'intuition suivante : "sachant que la simplification des règles de C4.5 rules semble obtenir de bonnes performances, quelles seraient son comportement si on l'appliquait sur la base de règles agrégée issue de l'apprentissage répété?"

On peut penser que partant d'une base plus riche, l'exploration de l'espace de recherche permettrait de dégager un meilleur prédicteur qui, sans surpasser le prédicteur agrégé, s'en approcherait, du moins serait supérieur en moyenne au classifieur initial.

Nous avons monté la procédure de cross-validation suivante :

1. on a constitué 10 paquets de chaque fichier benchmark,
2. sur les 9/10 des paquets nous avons construit les quatre classifieurs
   - SIPINA [Zighed *et al.*, 1992] avec une règle d'arrêt (test du χ^2 à 1%), ce sera notre classifieur de référence;
   - le classifieur précédent simplifié à l'aide de C4.5 rules [Quinlan, 1993a],
   - un classifieur agrégé "bagging" de SIPINA avec 25 répétitions,
   - une simplification du classifieur bagging en s'appuyant sur l'échantillon d'apprentissage;
3. comme dans toute cross-validation, on fait tourner les paquets pour obtenir 10 estimations de la moyenne et de la variance du taux de bons classements en validation et du nombre de propositions (entre parenthèses) dans la base de règles.

Les résultats sont consignés dans le tableau (10.1), pour une meilleure lecture nous n'y avons noté que les valeurs moyennes, les différences significatives par rapport à notre référence sont signalés par un signe (+) s'il est meilleur (la précision en généralisation est meilleure), (-) sinon.

On ne peut pas dire que ce soit un franc succès en ce qui concerne la simplification des bases de règles agrégées. Le tableau 10.1 amène néanmoins quelques réflexions intéressantes :

- la simplification d'une base de règles est effective sur l'arbre initial, elle réduit de manière drastique la complexité sans perte sur la précision;
- le bagging améliore en général les performances du classifieur sauf lorsqu'il est très stable, ce qui arrive lorsque la taille de la base initiale est faible. Dans deux cas seulement, il détériore les performances : sur le fichier Iris, c'est assez compréhensible, on sait que c'est un concept très facile à apprendre sur lequel la variance du classifieur est quasiment nulle; moins évidente en revanche est l'explication pour le fichier "pima-diabetes";
- enfin, la simplification de la base agrégée amène certes une diminution drastique de la base, le nombre de propositions est quasiment au même niveau que celui du modèle initial, en revanche les performances se dégradent.

Pour mieux comprendre ce phénomène, nous devons expliciter plus en profondeur la simplification de la base agrégée. Sur chaque fichier d'apprentissage, le bootstrap engage à chaque passage environ 63.2% des individus pour créer les règles. Lorsque la base complète est constituée par union des s classifieurs, les distributions de classes associées à chaque règle sont reéstimées

	SIPINA	**SIPINA Simplifié**	**Bagging**	**Bagging Simplifié**
autos	0.52 (34)	0.52 (27)	[+] 0.65 (1102)	[+] 0.56 (144)
breast	0.97 (77)	0.97 (26)	[+] 0.99 (1855)	0.96 (771)
car	0.88 (21)	0.88 (9)	0.89 (500)	[-] 0.82 (7)
machine	0.88 (7)	0.88 (7)	0.88 (179)	[-] 0.84 (9)
credit	0.96 (13)	0.95 (9)	0.96 (713)	0.95 (62)
flags	0.66 (15)	[-] 0.62 (10)	[+] 0.69 (494)	[-] 0.61 (43)
hepatitis	0.87 (20)	0.88 (6)	[-] 0.83 (457)	[-] 0.83 (6)
ionosphere	0.86 (70)	[+] 0.88 (9)	[+] 0.92 (1077)	[-] 0.83 (15)
iris	0.94 (14)	0.94 (11)	[-] 0.92 (263)	0.87 (15)
lung-cancer	0.44 (3)	0.44 (3)	[+] 0.56 (141)	[-] 0.22 (18)
pima-diabetes	0.77 (82)	0.78 (27)	[-] 0.75 (5227)	[-] 0.74 (35)
vote	0.95 (3)	0.95 (3)	0.95 (278)	0.96 (3)
wave	0.72 (108)	[-] 0.69 (39)	[+] 0.79 (3004)	[-] 0.63 (71)
wine	0.86 (24)	[+] 0.90 (18)	[+] 0.98 (465)	[+] 0.94 (11)
zoo	0.70 (9)	0.70 (8)	0.70 (210)	[+] 0.72 (8)

TAB. 10.1 – *Taux de bon classement et nombre de règles moyens pour une 10-fold cross-validation*

sur la totalité du fichier d'apprentissage afin de procéder à la simplification préconisée dans C4.5 [Quinlan, 1993a]. Il semble que ce soit ce passage qui pose problème. La pureté des sous-groupes diminue forcément, les règles prises individuellement dans la base agrégée que l'on propose à l'algorithme de simplification ne sont pas d'aussi bonnes qualités que si l'on avait optimisé l'apprentissage sur la totalité du fichier. De fait, il apparaît clairement que c'est surtout le vote qui fait la force du bagging et dans une très moindre mesure la qualité des prédicteurs qui le compose.

10.5.2 Les arbres sont de nouveau nés

Dans la monographie CART, [Breiman *et al.*, 1984] ont mis en exergue une propriété importante des arbres qui n'a que peu d'effet dans la pratique, du moins jusqu'à maintenant : si l'effectif d'apprentissage est infini et que la taille de l'arbre peut augmenter à souhait i.e produire des rectangles de taille micronesque dans un espace continu, alors le prédicteur qui travaille virtuellement sur des probabilités converge vers le prédicteur bayésien.

Bien entendu, cela est impossible dans la pratique, l'expansion de l'arbre entraîne une fragmentation croissante des échantillons et donc des estimations des distributions de probabilités de moins en moins assurées dans les parties basses de la structure. Conscients de cette limite, [Shang et Breiman, 1996] ont proposé une première méthode où l'on estime la distribution multidimensionnelle des observations avant de construire le classifieur en utilisant le générateur consti-

tué par les fonctions de répartition estimées. Hélas, ces estimations sont hautement paramétriques et sont discutables quant à leur vraie robustesse. Mais, c'était là ouvrir une brèche qui a donné jour à une idée originale[59] et enthousiasmante au point que les auteurs [Breiman et Shang, 1996] ont donné un titre assez poétique à leur papier "Born again tree".

Le principe fondateur est relativement simple : on utilise comme générateur de données le prédicteur agrégé construit par boosting M_A. On dispose ainsi d'un échantillon de taille virtuellement infinie sur lequel l'arbre construit devrait retrouver le taux de succès obtenu par M_A. Dans la pratique, afin de limiter la masse des calculs, les auteurs préconisent la génération d'un effectif suffisant sur chaque sommet afin de donner une meilleure assise au choix de la variable de segmentation. De fait, le succès de l'algorithme repose sur quatre étapes successives.

Création du classifieur agrégé

A partir de l'échantillon d'apprentissage Ω^a, on construit par apprentissage répété le classifieur agrégé M_A. De fait, pour tout individu ω de la population, nous pouvons lui associer une classe prédite $\widehat{y} = M_A(\omega)$. L'objet de l'étude est de construire à partir des données que l'on peut générer à l'aide de M_A, un arbre unique qui possède les même qualités en performances.

Génération des données

La solution la plus évidente est d'estimer la fonction de distribution multi-dimensionnelle des $(X_1(\omega), \ldots, X_p(\omega))$ puis de procéder à un échantillonnage à partir de cette distribution. [Breiman et Shang, 1996] proposent une solution plus simple qui permet de garder "une certaine corrélation entre les $X_i()$", et surtout qui soit adaptée que l'on ait des données numériques ou qualitatives.

Soit θ un seuil ($0 \leq \theta \leq 1$, couramment $\theta = 0.25$ ou 0.5) fixé par l'utilisateur. Pour une nouvelle observation artificielle ω', on sélectionne au hasard un individu $\omega \in \Omega^a$, pour chaque attribut $X_i(.)$ on génère un nombre η selon une loi uniforme de paramètre $(0, 1)$: si $\eta > \theta$ alors $X_i(\omega') = X_i(\omega)$ sinon on sélectionne une valeur au hasard parmi les $X_i(\Omega^a) = \{X_i(\omega) \,/\, \omega \in \Omega^a\}$. La classe à prédire sera tout simplement $Y(\omega') = M_A(\omega')$.

Construction de l'arbre

Plutôt que de générer un effectif initial titanesque, les auteurs préfèrent assurer la présence minimum de ns individus sur chaque sommet avant de segmenter. Concrètement, on générera autant d'individus qu'il faut que l'on glissera à partir du sommet initial jusqu'à ce que sur le

59. originale dans le sens où l'on utilise le prédicteur agrégé pour générer les données, [Craven et Schavlik, 1996] ont utilisé le même principe mais en générant les données à partir d'un réseau de neurones.

sommet considéré l'on dispose de ns observations. On peut fixer $ns = n$ mais il ne semble pas que cela influence beaucoup l'algorithme.

Taille optimale

Afin de déterminer la taille optimale, une première règle d'arrêt fondée sur la nature des données utilisées est mise en oeuvre : si aucun des noeuds enfants d'une segmentation ne contient un individu référencé dans Ω^a, on arrête le processus. En effet, cela donnerait un caractère trop artificiel à l'arbre construit, qui ne reposerait plus du tout sur des observations concrètes.

La deuxième méthode est sans surprise de la part d'un des auteurs de CART [Breiman *et al.*, 1984], c'est bien sûr l'élagage. Comment dans l'étape précédente, on générera les données qui serviront d'échantillon d'élagage.

Au final, [Breiman et Shang, 1996] se rendent compte que l'arbre résultant de ce processus est bien supérieur, sur les données benchmark, que la méthode CART classique; en revanche ses performances sont inférieures à celles du prédicteur agrégé. Malgré la quantité d'observations illimitée, l'arbre reste assez éloigné de M_A, qui sert de référence (prédicteur optimal) ici puisqu'il est impossible de faire mieux dans la mesure où les données traduisent ce concept. En conclusion de leur papier, les auteurs pensent que l'on peut encore améliorer le système de génération des observations, mais que de toute manière on sera toujours limité par le système de représentation utilisé. Nous rejoignons cette dernière remarque car il semble en effet que pour les données de type continu, le vote dans le classifieur agrégé répond de manière adéquate à l'incertitude liée à la détermination des bornes de discrétisation. En passant de nouveau à des arbres durs [60], nous perdons cette spécificité. D'ailleurs dans cette optique, certains auteurs pensent que les découpages flous, traduisant l'incertitude autour de la vraie valeur de la borne, réduisent considérablement la variance dans les arbres de décision [Dietterich et Kong, 1995b].

10.6 Conclusion

La construction de prédicteurs agrégés dans les graphes d'induction est un champ de recherche très prolifique qui connaît un grand regain d'intérêt depuis les récents travaux de [Breiman, 1994] [Breiman, 1996a], la plupart des papiers ne sont encore disponibles que sous la forme de rapports techniques. Des auteurs célèbres comme [de Merckt et Quinlan, 1996] pensent même qu'il s'agit là d'un des centres d'intérêts les plus importants du futur. Pour notre part, il est clair qu'une des conséquences les plus passionnantes des travaux autour de l'agrégation des classifieurs est d'une part d'avoir mis le doigt sur une des faiblesses essentielles des graphes, à savoir leur

60. "crisp"

grande variance, d'autre part d'avoir fourni les outils d'analyse qui permet d'appréhender et de solutionner ce problème.

D'un autre côté, même si nous notons que l'augmentation des performances est indéniable dans la plupart des expérimentations que nous avons consultées, y compris les nôtres, nous restons réservés quand même sur cette stratégie qui fait perdre aux graphes une de leurs spécificités les plus séduisantes : la lisibilité du classifieur.

Au final, il nous apparaît que la véritable solution passe par une meilleure absorption de la variance : soit par la structure de l'arbre (arbres flous, arbres à options), soit par la nature des données utilisées (augmentation artificielle des effectifs, heuristiques de suppression des données bruitées).

Quatrième partie

Réalisation logicielle et applications

Chapitre 11

Une plate-forme d'ingénierie des connaissances : le logiciel SIPINA_W©

11.1 Introduction

Un des principaux attraits de faire une thèse en apprentissage est la possibilité de tester expérimentalement la viabilité des nouvelles idées en appliquant l'algorithme sur des bases de données exemples, artificielles ou réelles. Dans ce cadre, il n'est pas rare que l'on voie émerger, dans les différents travaux que nous avons consultés, des programmes informatiques ad-hoc, souvent avec une interface sommaire, ayant pour objectif de montrer le fonctionnement effectif de la méthode et ses possibilités en matière d'apprentissage. Ces logiciels satisfont leurs auteurs dans le cadre de leur travail mais sont très rarement utilisables par d'autres, ne serait-ce que pour vérifier les résultats obtenus.

Nous avons voulu aller plus loin dans notre travail en proposant un système complet d'ingénierie des connaissances dont l'objectif est double. Le premier est de servir de base de travail pour l'évaluation et la caractérisation des algorithmes d'induction, dans cette thèse comme dans d'autres [Rabaseda, 1996]. Quelques auteurs dans la littérature ont adopté la même démarche en proposant des systèmes complets de constructions d'inductions par graphes, les plus célèbres sont l'oeuvre d'auteurs illustres tels que [Buntine et Caruana, 1991] avec le système IND, ou encore [Kohavi *et al.*, 1994] avec la bibliothèque MLC++. La plupart de ces outils sont conçus pour fonctionner sur des stations de travail, avec le système d'exploitation UNIX. Même si l'on dispose du code source, l'absence d'interface digne de ce nom les confine à une exploitation expérimentale, idéale pour les chercheurs mais impraticable pour un utilisateur non averti.

Le second objectif est de créer un produit suffisamment fini pour que sa mise en oeuvre dans le milieu des entreprises (industriel, médical, administration...) soit possible sans aucune adaptation. Dans cette optique, le choix d'un progiciel fonctionnant sous le système Windows®

n'est pas innocent. Malgré toutes les réserves que l'on peut émettre sur l'efficacité de ce système d'exploitation, il est certainement le plus répandu dans le monde des entreprises depuis l'explosion de la micro-informatique et la diffusion de plus en plus étendue des ordinateurs compatibles PC. Certes il existe de nombreux logiciels commerciaux fondés sur le paradigme des arbres de décision, nous nous en démarquons cependant car, plutôt que de proposer un outil confiné à la description ou à la structuration simple de données, nous avons utilisé explicitement le cadre général de l'extraction de connaissances à partir des données[61] lors de la spécification de l'architecture du logiciel. Nos repères ont été les principales étapes de l'ECD que nous avons détaillées dans le chapitre introductif, à savoir : l'acquisition et la sélection des données, le pré-traitement des observations pour l'extraction de connaissances proprement dite réalisée à l'aide des algorithmes d'induction par graphes, l'évaluation statistique et empirique des formes extraites, et la mise en forme des connaissances en vue de leur exploitation soit pour l'explication des formes découvertes, soit pour la généralisation dans la population étudiée.

Le logiciel SIPINA_W© est issu d'une longue évolution sous la direction et l'instigation du Pr. Zighed. La première version, écrite en Pascal sous DOS, prenait appui sur les résultats théoriques mis en exergue dans la thèse de [Zighed, 1985]. Le système à l'époque ne prenait en compte que les attributs catégoriels. Un premier saut qualitatif fût accompli avec la version livrée à l'intérieur de l'ouvrage "SIPINA : Méthode et logiciels" de [Zighed *et al.*, 1992]. Le logiciel pouvait traiter les attributs continus, que l'on rencontre souvent dans les problèmes réels, en introduisant le processus de discrétisation. A cette époque, compilé sous le système DOS, il fonctionnait en mode réel avec une limite de taille mémoire à 640 Ko, insuffisante pour traiter les grosses bases de données, courantes dans les applications d'ECD. Une seconde étape importante a été le passage sous le système Windows®. A partir d'une plate-forme générique d'induction de graphes comprenant un éditeur de données et une interface graphique, mes principales contributions ont été de trois ordres : greffer les autres méthodes d'induction par graphes (C4.5 [Quinlan, 1993a], CART [Breiman *et al.*, 1984]...) ; mettre en place les dispositifs nécessaires aux fins de comparaisons, d'évaluation et de sélection de modèles (cross-validation, bootstrap, apprentissage répété....) ; intégration des derniers développements issus de la recherche, propres à mes travaux (extraction sélective des règles par validation [Rakotomalala et Chettouh, 1996], détection automatique de taille minimale des sommets [Rakotomalala *et al.*, 1996], discrétisation [Zighed *et al.*, 1997]...) ou issus de travaux d'autres chercheurs (agrégation de classifieurs [Breiman, 1996a], simplification des règles [Quinlan, 1993a]...).

La diffusion du logiciel SIPINA_W© sur Internet nous a permis de l'améliorer constamment. Nous avons essayé de mettre à profit les critiques, remarques et suggestions en provenance d'utilisateurs du monde entier, pour la plupart d'origine anglo-saxonne, oeuvrant dans des domaines aussi divers que la recherche en apprentissage, la médecine, la géologie, ou encore l'assurance.

61. Knowledge Discovery in Databases

Dans ce chapitre, nous présentons l'architecture globale du logiciel SIPINA_W©. Nous mettrons en exergue les différents modules constitutifs qui s'inscrivent dans le cadre de l'extraction automatique de données par apprentissage supervisé. Bien entendu, rien n'est définitif, la plate-forme actuelle subit constamment des modifications dans le sens d'une amélioration des performances et de la souplesse de l'outil. Plutôt que de fournir une description approfondie des options associées aux nombreux menus du logiciel, disponible dans le guide de l'utilisateur [Zighed et Rakotomalala, 1996b], nous préférons présenter son esprit et les réalisations afférentes, passées ou parfois à venir.

11.2 Implémentation et élaboration de SIPINA_W©

Les exigences initiales à l'origine de la construction du logiciel d'induction par graphes sous l'environnement Windows® ont été les suivantes :

1. une interface graphique aux normes Windows® (figure 11.1);
2. une visualisation en temps réel de l'induction avec possibilité pour l'expert d'agir de manière interactive pour modifier la configuration du graphe, soit en choisissant les variables de segmentation, soit en choisissant une opération particulière (fusion, segmentation, limitation du nombre de noeuds). En ce sens, il est important de proposer de nombreux outils d'aide à la décision;
3. des capacités de traitement élevées. Théoriquement, il est de $(2^{32}-1)$ observations et 16384 variables, dans la réalité la taille de la mémoire virtuelle[62] est la limite que l'on ne peut dépasser.

Le langage d'implémentation est le Pascal Objet, la première version qui ne fût jamais diffusée a été écrite sous l'environnement Borland Pascal 7.0. Le choix de ce langage tient à la fois de la culture propre à l'enseignement en informatique à l'Université Lumière Lyon 2 et des impératifs de cohérence inhérents aux projets de cette ampleur. En effet, plusieurs personnes ont travaillé simultanément. Atteindre un certain niveau de modularité, rendue aisée par l'adoption de la technique objet, était nécessaire pour assurer une coordination efficace des différentes unités. Par rapport à des langages plus flexibles tels que le C++ (et peut être plus puissant bien que cela reste à prouver, le compilateur Borland du langage Pascal ayant constamment progressé ces dernières années), le Pascal propose une très riche bibliothèque de types et outils prédéfinis, que l'on peut enrichir et modifier à souhait.

Le choix du langage Pascal s'est d'autant plus révélé judicieux que l'environnement de programmation visuelle Delphi a fait son apparition au printemps 1995. De fait, nous avons pu

62. Elle dépend à la fois de la mémoire RAM et de l'espace disque disponible.

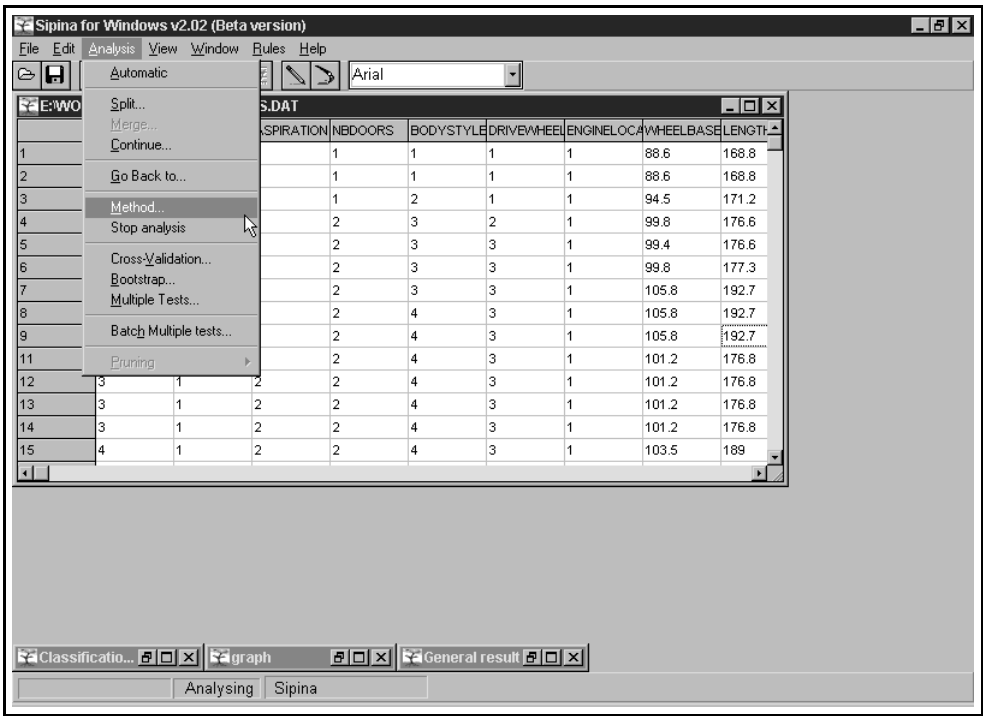

FIG. 11.1 – *Interface de base du logiciel SIPINA-W avec son menu principal*

réduire de moitié la taille du code source en supprimant toutes les parties relatives à la programmation de l'interface. A l'heure actuelle, SIPINA_W© comporte près de 50000 lignes de codes, la mise à jour du logiciel par intégration de nouvelles fonctionnalités et/ou méthodes est relativement aisée. Le prochain enjeu est maintenant le passage de la version 16 bits à une version 32 bits, exploitant directement les nouvelles possibilités des systèmes d'exploitation 32 bits. Notons que la plate-forme actuelle fonctionne sur n'importe quel système Windows® (3.1, 3.11, 95, NT) pourvu qu'il tourne en mode étendu.

11.3 Architecture globale de SIPINA_W© dans le cadre de l'ECD

Notre propos est donc de développer un système complet incluant les étapes de l'extraction automatique de connaissances à partir de données qui permettront à l'utilisateur final : d'une part, d'analyser les causalités mises à jour; d'autre part, de proposer un diagnostic aussi fiable que possible, ou du moins dont on connaît la propension à l'erreur. Les modules composant le logiciel sont résumés dans le graphique 11.2. Dans les sections suivantes, nous aborderons tour à tour les éléments qui les composent.

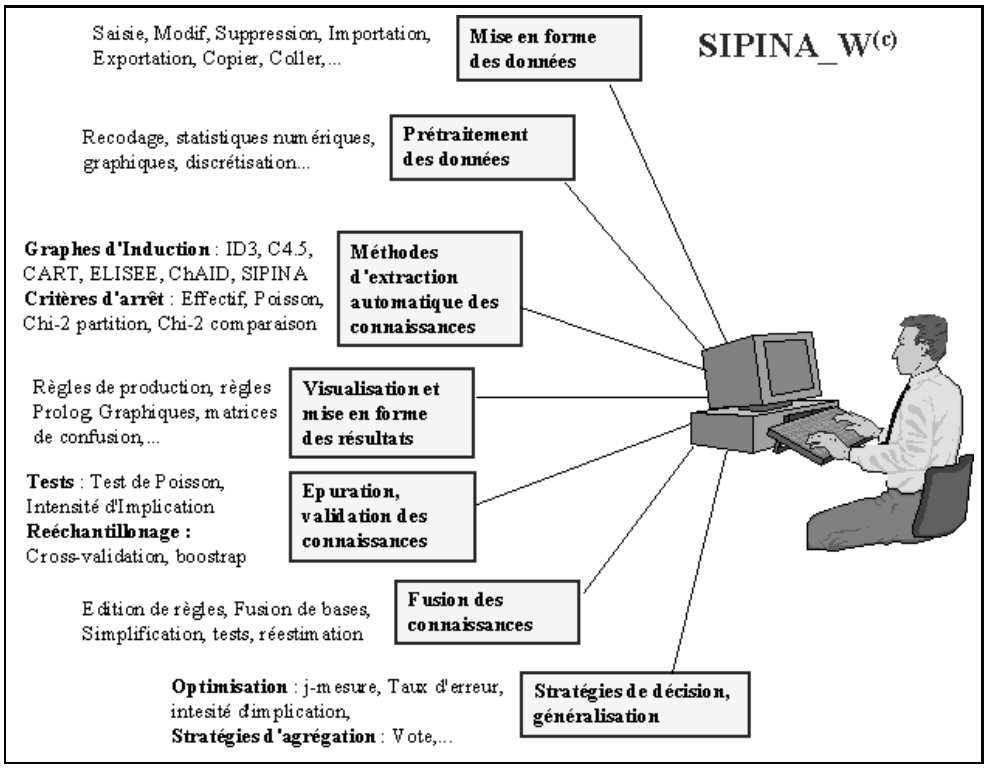

FIG. 11.2 – *Architecture du logiciel SIPINA-W(c)*

11.3.1 Mise en forme des données

L'acquisition des données comporte une première phase qui consiste à circonscrire au mieux le domaine d'étude afin que les individus étudiés soient aussi homogènes que possible, ce travail revient essentiellement à l'expert. L'étape suivante est l'extraction d'une collection d'observations : dans le cadre de l'ECD elle a souvent pour origine une requête sur un serveur de données, dans le cas général il peut provenir de n'importe quelle source. En tous les cas, nous devons disposer d'une matrice de données avec en ligne les observations, en colonne les attributs qui les décrivent, y compris la variable à prédire.

Le format de fichier SIPINA_W© est un format propriétaire hérité des précédentes versions du logiciel, notre rôle a consisté en grande partie à construire des convertisseurs qui permettent d'importer d'autres formats de fichiers (figure 11.3) dont les principaux sont : tableurs (Lotus), base de données (Paradox, dBase), éditeurs de texte (fichiers texte avec séparateurs), ou encore serveurs de données benchmark comme le fameux UCI Irvine [Murphy et Aha, 1995]. Nous sommes en train actuellement de travailler sur les opportunités de produire des données tabulaires directement à partir de vues produites par des requêtes sur des serveurs de base de données. Compte tenu de la puissance de l'environnement de programmation Delphi, il est à prévoir que l'intégration de cette option sera relativement aisée. Bien entendu, le pendant de l'importation,

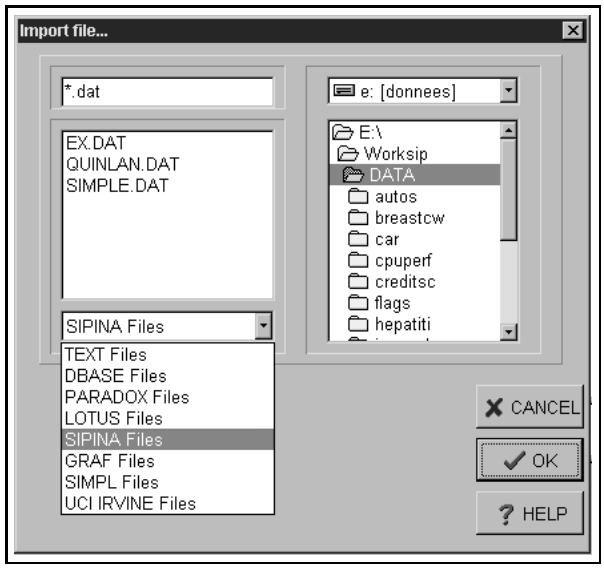

FIG. 11.3 – *Boîte de dialogue d'importation de données dans DataManager*

l'exportation, a été également implémenté.

Cette étape, couplée avec le pré-traitement des données que nous aborderons plus loin, est tellement importante que nous avons préféré construire un module indépendant consacré uniquement aux tâches de manipulations de données. Le logiciel associé, dénommé *DataManager*, a été écrit en collaboration avec Mlle Valérie Goyet, alors stagiaire au sein du Laboratoire ERIC.

11.3.2 Pré-traitement des données

C'est une étape fondamentale de l'ECD. Souvent les données sont entâchées d'imperfections (données manquantes, attributs non-pertinents...). De plus, certaines méthodes d'extraction de connaissances exigent qu'elles soient présentées d'une manière particulière (données de type continu et distribution normale pour l'analyse discriminante, données de type catégoriel pour les algorithmes symboliques...). Il importe alors de développer des stratégies qui nous mènent vers la meilleure adéquation entre l'algorithme d'apprentissage et les observations en entrée tout en contrôlant les approximations introduites par ce pré-traitement.

Certaines des fonctions présentées dans cette sous-section sont en cours de développement dans le module DataManager, nous nous contenterons donc de décrire les principes qui sous-tendent notre travail dans les différents cas de figure.

Sélection des individus

Avec l'augmentation de la capacité de stockage des machines, la disponibilité des observations n'est plus un problème, c'est la situation inverse qui semble maintenant prédominer : il y a trop

de données disponibles. Dans ce contexte, les réactions sont diverses :

- développer des algorithmes spécifiques extrêmement rapides [Catlett, 1991a];
- développer des approches adaptées aux machines parallèles [Holsheimer *et al.*, 1996];
- développer enfin des techniques d'échantillonnage (ou de fenêtrage) qui permettent d'extraire le maximum d'informations avec le minimum de données [Quinlan, 1993a].

Pour ou contre l'échantillonnage, les avis sont partagés. Il est vrai que dans certains cas, les phénomènes sont suffisamment rares pour que son appréhension ne soit possible qu'à travers des bases de grande taille. Si nous considérons le cas des fraudes bancaires [Fortune *et al.*, 1996], il y a à peu près une transaction frauduleuse pour 10^5 opérations. Si l'on procède à un échantillonage aléatoire simple de 1000 individus, il y a de grandes chances que la classe "Fraude" ne soit tout simplement pas représentée.

Malgré tout, il nous semble vain d'essayer de traiter entièrement les grosses bases de données. Quelle que soit l'augmentation de la puissance des machines, il est peu probable qu'elle suive la croissance de la masse de données, il est sans aucun doute plus profitable de mettre en oeuvre d'autres schémas de tirage tout en adaptant les algorithmes d'apprentissage en conséquence. Fort heureusement, les graphes sont équipés pour répondre de manière adéquate à ce type de problème [Breiman *et al.*, 1984], notre rôle consistera alors à implémenter les différentes méthodes d'échantillonnage (stratifiés, méthode des quotas...) que l'utilisateur peut demander.

Sélection des variables

Les graphes d'induction sont réputés pour sélectionner uniquement les variables pertinentes en induction grâce aux mesures d'évaluation des segmentations. Ceci est moins vrai lorsqu'il s'agit de stratégies d'induction par arbres qui utilisent le principe de l'élagage. La construction est dite "hurdling", une variable non-pertinente est introduite même si elle n'apporte pas significativement d'informations; l'objectif sous-jacent est bien sûr, comme on a pu le montrer dans cette thèse, l'appréhension des concepts pour lesquels certaines variables inter-agissent.

Dans cette optique, la sélection des variables devient cruciale car elle évite l'insertion d'attributs non-pertinents dans le classifieur. Ces dernières années de nombreux auteurs se sont attentivement penchés sur ce problème. Deux écoles s'affrontent : ceux qui préconisent la sélection des variables hors algorithme d'apprentissage [Almuallim et Dietterich, 1991] [Almuallim et Dietterich, 1992] [Rauber et Steiger-Garcao, 1993] [Sebban, 1996], et ceux qui préconisent d'utiliser explicitement l'algorithme d'apprentissage [Kohavi et John, 1997] [Cherkauer et Shavlik, 1996].

La seconde méthode est peut-être plus efficace, mais elle est très coûteuse en temps de calcul. En effet relancer plusieurs fois l'algorithme d'apprentissage pour sélectionner le bon ensemble

de variables entraîne une multiplication des coûts qui peut s'avérer préjudiciable. De plus, cette méthode ne s'inscrit pas dans le cadre de notre propos ici.

La première méthode, une sélection ad-hoc des variables avant l'application de l'algorithme, paraît cependant discutable en reconnaissance de formes. Rien ne laisse présager qu'une variable sélectionnée d'une manière quelconque puisse être pertinent par la suite dans un algorithme d'apprentissage s'appuyant sur un système de représentation des connaissances particulier [Imam, 1994]. Il reste néanmoins dans la pratique que la réduction de la dimension est nécessaire tant les graphes d'induction peuvent être sensibles aux attributs non-pertinents dans certains cas, notamment lorsqu'il y a interaction entre plusieurs variables [Kira et Rendell, 1992] [Langley et Sage, 1994].

Recodage et transformation des données

Ces transformations sont parfois considérées comme des méthodes d'appauvrissement ou d'enrichissement des données, dans ce dernier cas l'opération est rendue possible par l'adjonction d'hypothèses supplémentaires. Les principaux objectifs sont l'homogénéisation des données et l'adjonction de nouvelles propriétés de manière à pouvoir appliquer directement les différents algorithmes d'apprentissage.

Selon la nature des données, nous distinguons différentes transformations que nous avons implémentées dans SIPINA_W© et DataManager (figure 11.4):

1. attributs catégoriels : une technique fréquemment utilisée est le codage disjonctif complet, en fait nous retrouvons ici les méthodes introduites dans la partie sur le regroupement des valeurs des attributs.

2. attributs continus : dans le cadre de l'induction par graphes, une transformation qui nous tient à coeur est la discrétisation des attributs, dans la phase préparatoire seul le découpage en intervalles multiples (soit fixé par l'utilisateur, soit déterminé par la méthode) est véritablement justifié.

Traitement des données manquantes

Le problème des données manquantes peut être considéré sous différents angles dans le classement par graphes. Le premier, que nous n'aborderons pas ici, se situe dans la phase de généralisation [Quinlan, 1987b] [Brunet, 1992]. Le second survient en apprentissage, un ou plusieurs individus possèdent des attributs non référencés.

La solution la plus simple consiste à supprimer les observations correspondantes. Il est clair qu'une telle politique n'est pas très appropriée dans le traitement sur bases réelles. L'exclusion

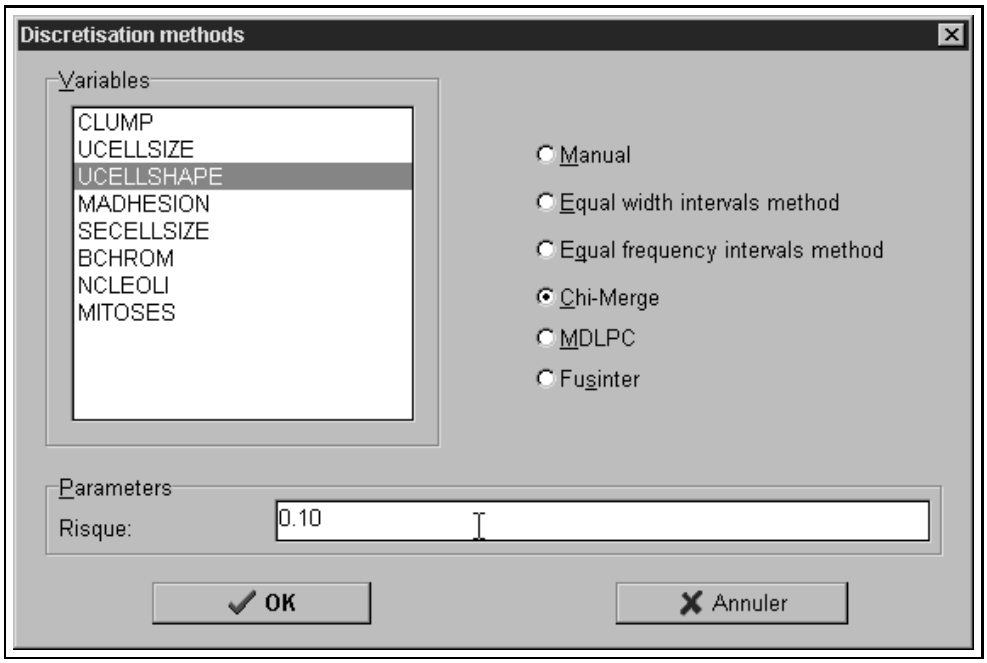

FIG. 11.4 – *Boîte de dialogue de sélection des méthodes de discrétisation dans une phase de recodage des attributs continus*

d'individus occasionne des pertes d'informations, d'autant plus regrettables que parfois les observations manquantes portent sur des variables non discriminantes[63] en apprentissage. Les autres solutions, autrement plus efficaces, sont regroupées en deux familles distinctes :

- soit nous traitons directement l'absence de données dans l'induction, [Quinlan, 1989] recense les différentes méthodes utilisées en ce sens, l'objectif étant de construire le classifieur le plus performant en généralisation;

- soit nous cherchons à reconstruire les données originelles par différentes méthodes mono ou multivariées. Pour les données continues nous pouvons envisager l'utilisation de la moyenne, la médiane, ou encore la régression; pour les données catégorielles, nous pouvons nous servir de la valeur la plus fréquente, de l'analyse bivariée sur un tableau de contingence, ou encore de la méthode des plus proches voisins. L'objectif dans ce cas est de retrouver la bonne valeur relative à l'individu concerné.

Nous penchons pour le deuxième type d'approche, elle est générale à tout type de processus d'apprentissage. Sur *DataManager*, nous travaillons actuellement à implémenter les stratégies appropriées afin de restituer les valeurs originelles des cases vides dans la matrice de données.

63. cet argument milite encore en faveur de la sélection des variables comme pré-traitement

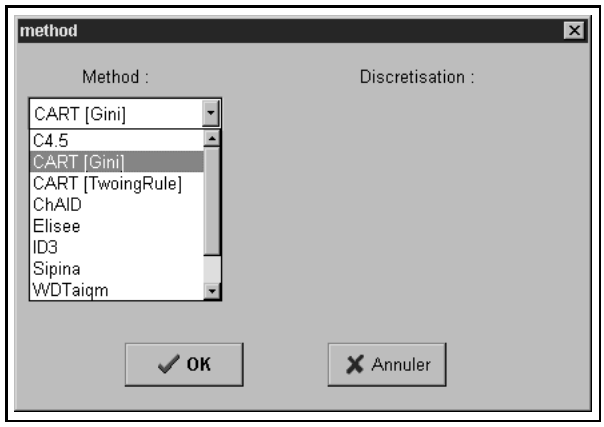

FIG. 11.5 – *Sélection des méthodes dans SIPINA-W(c)*

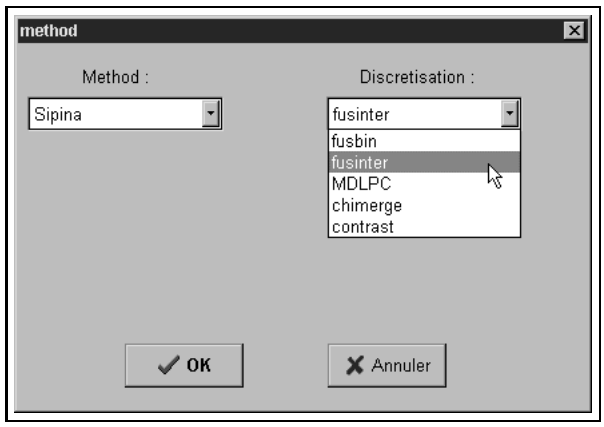

FIG. 11.6 – *Sélection de la discrétisation locale dans SIPINA-W(c)*

11.3.3 Extraction de connaissances

C'est le point d'orgue du logiciel. Tout le long de notre thèse, nous nous sommes attachés à comprendre et à tester les différentes méthodes rapportées dans la littérature. Finalement, les algorithmes d'induction par graphes diffèrent peu dans leur principe, à partir d'une base simple intégrant la segmentation et le regroupement de noeuds, nous avons pu couvrir l'ensemble des stratégies. Les véritables points de différence sont la recherche de la taille optimale qui peut être réalisée grâce à l'adoption d'une règle d'arrêt (les différents critères peuvent être mis en oeuvre sur n'importe quel algorithme) ou grâce à l'élagage (par optimisation ou par estimation du taux d'erreur).

Dans la plate-forme SIPINA_W©, ces algorithmes sont accessibles via une boîte de dialogue où l'on peut choisir la méthode appropriée (figure 11.5). Pour la méthode d'induction de graphes Sipina [Zighed et Rakotomalala, 1996a], nous disposons d'une pléiade de stratégies de discrétisation binaire ou n-aire (figure 11.6). Avec ces différentes combinaisons, nous pouvons tester plusieurs possibilités pour choisir la méthode la plus appropriée sur un domaine donné, en

utilisant les outils de comparaisons que nous détaillerons plus loin.

Aux fins de vérifications et de comparaisons, nous avons beaucoup travaillé sur une implémentation fidèle des algorithmes d'induction par graphes disponibles dans la littérature. Nous détaillons dans cette section les stratégies que nous avons implémentées, nous insisterons surtout sur les références bibliographiques utilisées.

Les méthodes d'induction par graphes implémentées dans SIPINA_W©

Ce sont les méthodes de base, comportant des heuristiques propriétaires. Certaines sont des références incontournables, nous les avons testées sur les bases exemples utilisées dans cette thèse (voir Annexes) :

- **C4.5** : on peut trouver sur le site WEB de l'auteur les codes sources des dernières améliorations de cette méthode qui figure parmi les références reconnues de la littérature. Nous nous en sommes tenus aux procédures décrites dans l'ouvrage de [Quinlan, 1993a]. Notons qu'il existe actuellement une version commerciale C5.0 intégrant les dernières "trouvailles" de l'auteur sus-cité.

- **CART** : deuxième incontournable de la littérature, nous avons programmé les deux versions de cette approche [Breiman *et al.*, 1984]. Notons que notre implémentation de la méthode *Twoing* ne prend en compte que les variables continues, celle de la *Gini* est une extension n-aire.

- **ChAID** : c'est une transcription fidèle de la méthode de [Kass, 1980], on peut la retrouver dans des logiciels commerciaux tels que Knowledge Seeker© d'Angoss Software.

- **Elisee** : très proche de la méthode Twoing de CART, elle utilise une distance du χ^2 et produit des arbres binaires. Cette méthode, telle que nous l'avons programmée, ne fonctionne correctement que sur des attributs prédictifs continus. Le papier de référence est celui de [Bouroche et Tenenhaus, 1970].

- **ID3** : c'est la méthode certainement la plus citée dans la littérature en intelligence artificielle, nous nous sommes servis de la stratégie de base [Quinlan, 1979], couplée avec la règle d'arrêt basée sur un test d'indépendance nous retrouvons l'algorithme décrit dans [Quinlan, 1986b].

- **SIPINA** : c'était notre point de départ, la stratégie pour laquelle nous avons développé la plate-forme. Avec le recul, nous nous rendons compte que ce fût un choix heureux car étant une généralisation des arbres elle englobe ainsi quasiment toutes les méthodes d'induction par graphes. Les documents de référence sont [Zighed, 1985], [Zighed *et al.*, 1992] et [Zighed et Rakotomalala, 1996a]. Notons que nous pouvons démultiplier les variantes en

choisissant des procédures de discrétisation différentes : FUSINTER[64] [Zighed *et al.*, 1996], MDLPC [Fayyad et Irani, 1993], CHI-MERGE [Kerber, 1992] et CONTRAST [de Merckt, 1993]. A notre connaissance, SIPINA_W© est la seule plate-forme diffusée, avec le package IND de [Buntine et Caruana, 1991] qui implémente les graphes de [Oliver et Wallace, 1991], qui propose la généralisation des arbres de décision aux graphes d'induction.

- **WDTaiqm** : c'est une transcription fidèle du papier de [Wehenkel, 1993] qui, depuis un certain nombre d'années s'intéresse de près à l'analyse de la stabilité transitoire des réseaux électriques [Wehenkel et Pavella, 1991].

- **QR_MDL** : elle s'appuie sur la théorie de la description minimale décrite dans [Quinlan et Rivest, 1989], nous avons utilisé ici l'option arbre n-aire.

- **PRETree** : cette stratégie induit des arbres de décision en utilisant l'interprétation en terme de régression de l'induction par graphes. La méthode est détaillée dans le chapitre consacré à la "Détection de la taille optimale des graphes (suite)", la publication correspondante est [Rakotomalala et Zighed, 1997].

Les règles d'arrêt d'expansion du graphe

Lors de la phase d'expansion du graphe, nous avons vu dans cette thèse que diverses règles d'arrêt pouvaient limiter la taille du classifieur. Certaines sont classiques, que l'on retrouve dans des algorithmes standards, d'autres en revanche sont ad-hoc et correspondent à des tentatives propres à notre travail de recherche. Voici la liste de règles d'arrêt disponibles dans la plate-forme SIPINA_W© (figure 11.7):

- **taille d'un sommet** : comme condition d'acceptabilité elle permet de contrôler les sommets produits dans le graphe i.e si un au moins des sommets enfants produits par un éclatement comporte un effectif inférieur à cette limite, la partition est refusée. De fait, il ne peut y avoir de noeuds de taille insuffisante [Breiman *et al.*, 1984] [Zighed *et al.*, 1992].

- **le test de Poisson** : c'est un test fondé sur l'intensité d'implication de [Lerman *et al.*, 1981], on arrête le développement sur un noeud dès qu'il est possible d'extraire une classe conclusion au seuil fixé par l'utilisateur.

- **le test d'indépendance du χ^2** : on l'utilise comme condition d'acceptabilité d'un éclatement. Si l'indépendance entre l'attribut prédictif et la variable à prédire n'est pas démentie au seuil fixé, nous refuserons, à l'instar de [Kass, 1980] [Quinlan, 1986b], la segmentation.

64. FUSBIN est une variante binaire de FUSINTER

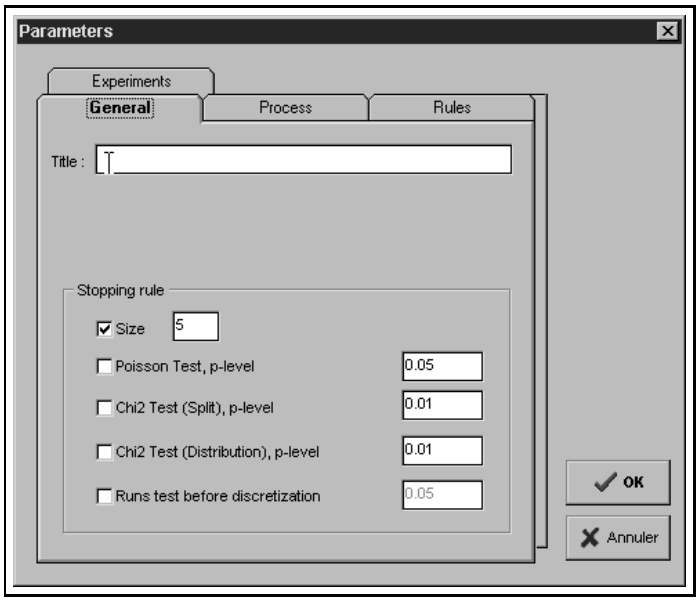

FIG. 11.7 – *Options de règles d'arrêt dans la plate-forme SIPINA-W(c)*

– **le test d'équivalence distributionnelle du χ^2** : ce test peut être vu comme une généralisation du test de Poisson ci-dessus. Un noeud est déclaré terminal dès que l'on observe une divergence significative de sa distribution des classes avec celle du sommet initial. On procède de manière analogue dans le premier algorithme CN2 [Clark et Niblett, 1989] pour stopper la spécialisation des règles en induction.

– **test des séquences, préalable à la discrétisation** : pour que l'on discrétise un attribut continu, il faudrait s'assurer de la séparabilité des individus. Nous avons beaucoup travaillé sur l'opportunité de l'introduction de ce test avant le découpage [Rabaseda *et al.*, 1995] [Rakotomalala, 1995b], il est apparu que sa puissance est du même niveau qu'une discrétisation classique où l'on refuse de segmenter lorsqu'elle aboutit à la construction d'un seul intervalle. Le test est d'autant moins intéressant que nous n'avons tabulé que les valeurs seuils pour un risque de première espèce de 5%, stocker les valeurs pour les autres risques entraînerait un encombrement mémoire (ou des accès disques si l'on décide de les lire dans un fichier) peu avantageux compte tenu de sa faible puissance.

Extraction et traitement des règles

Le passage du graphe aux règles est une étape importante de notre système d'apprentissage : d'une part, nous mettons la connaissance sous une forme exploitable, notamment par les systèmes à base de connaissances fonctionnant en Prolog ou utilisant des formats de règles natifs tels que le logiciel SelfMind (Système Expert en Logique Floue et Multi-valuée pour l'Imitation Naturelle des Décisions) développé au sein de notre laboratoire; d'autre part, ce changement de représentation

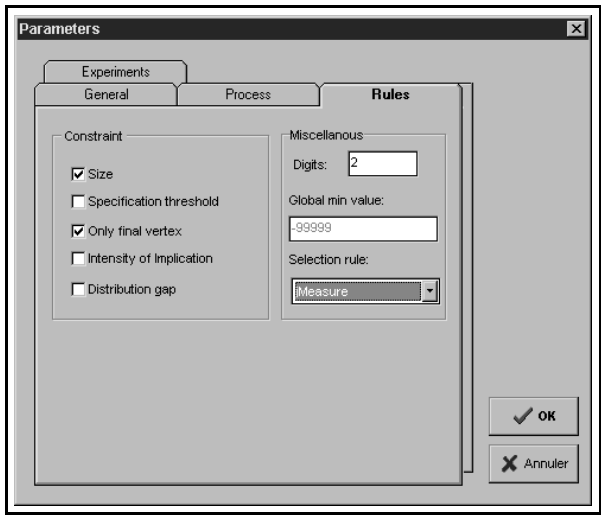

FIG. 11.8 – *Modes de génération des règles dans SIPINA-W(c)*

permet l'application de nouveaux opérateurs en vue d'améliorer les performances du classifieur (e.g. simplification, fusion avec d'autres bases de règles...).

Dans cette sous-section, nous détaillerons deux facettes du traitement des règles dans notre plate-forme d'ingénierie des connaissances :

1. stratégies et contraintes d'extraction des règles : les règles extraites du graphe ne sont certainement pas de la même qualité, certaines sont meilleures parce qu'à la fois plus précises et plus générales que d'autres. Sur la base des travaux de [Goodman et Smyth, 1988], nous avons mené une étude approfondie de l'influence de l'indicateur de qualité des règles sur les performances du classifieur induit [Rakotomalala *et al.*, 1997b]. Les résultats obtenus confirment la viabilité et l'avantage que l'on peut tirer d'un "bon" indicateur dans les caractéristiques de la base de règles construite. Sachant que chaque noeud non-initial du graphe constitue une règle potentielle, il apparaît plus opportun de les valider individuellement quitte à les mettre en concurrence pour définir le meilleur ensemble de règles extrait du graphe [Rakotomalala *et al.*, 1996] [Rakotomalala et Chettouh, 1996]. L'exclusion des règles redondantes peut alors être facilement résolue par un algorithme de simplification [Rabaseda *et al.*, 1996a]. Les options de génération de règles disponibles (figure 11.8) dans notre logiciel sont :

   – effectif minimal couvert par la règle : en-deçà d'un certain seuil, on suppose que la règle couvre trop peu d'individus pour qu'elle soit fiable. On ne peut pas dire par exemple que les malgaches sont myopes sous prétexte que l'auteur de cette thèse l'est.

   – seuil de spécialisation : si le nombre de contre-exemples à la règle est trop élevé, il est clair qu'elle est peu crédible. Le seuil de spécialisation indique la valeur limite du

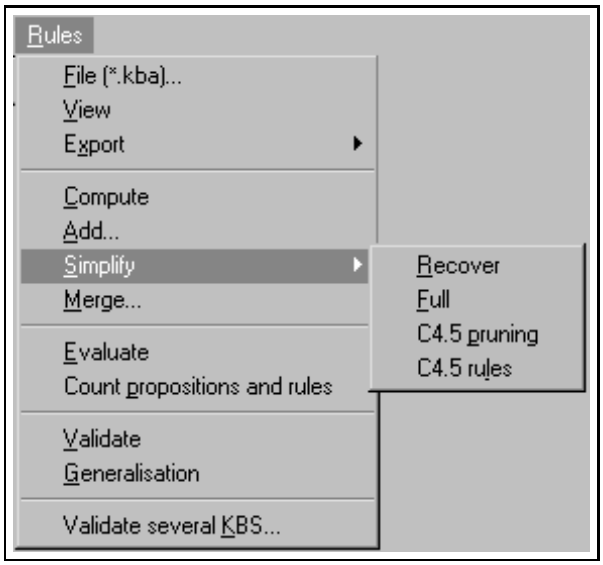

FIG. 11.9 – *Menu "Traitement des règles" dans Sipina-W(c)*

nombre de bon classements en apprentissage à partir duquel nous accordons notre confiance à la règle. Nous remarquerons qu'associés à la contrainte précédente, nous retrouvons réunis les critères de précision (peu de contre-exemples) et de généralité (beaucoup d'exemples couverts). Ce tandem a été utilisé par [Zighed, 1985] pour valider les règles dans les graphes bien avant les travaux de [Goodman et Smyth, 1988], ces derniers ont avant tout eu le mérite de proposer un indicateur synthétique en se basant sur l'explicitation de ces deux contraintes.

– extraction des règles sur les sommets terminaux : selon que cette option est cochée ou non, les règles seront extraites classiquement des feuilles ou de tous les noeuds non-initiaux [Rakotomalala et Zighed, 1996]. Pour que cette dernière stratégie soit efficace, elle doit être couplée avec l'option de validation et la simplification afin d'éliminer les règles peu fiables et/ou redondantes.

– validation des règles à l'aide de l'intensité d'implication : une règle pourra être validée à l'aide du test de causalité de [Lerman *et al.*, 1981]. On note que cette possibilité rend d'immense services dans des domaines où, plus que le classement à tout prix, il est préférable de ne pas conclure dès que la décision n'atteint pas un niveau de fiabilité suffisant. C'est le cas par exemple en médecine où les décisions à tort sont d'un coût très élevé.

– écart de distribution : c'est une extension fondée sur le test du χ^2 du test de causalité précédent, on veut s'assurer que les distributions de classes s'écartent suffisamment de la distribution initiale.

2. stratégies de simplification des règles : le traitement des règles tient une place prépondérante dans le logiciel SIPINA_W© (figure 11.9). Elle peuvent être modifiées, supprimées, ou encore réévaluées sur d'autres fichiers de données... Parmi les différentes opérations possibles, la simplification est assez particulière. Elle répond à deux impératifs : une diminution de la complexité du classifieur (estimée par le nombre moyen de propositions dans les prémisses et/ou le nombre de règles dans la base) et une augmentation concomittante des performances en généralisation. Si le bien fondé de cette dernière propriété fait l'objet d'âpres discussions, tous les chercheurs s'accordent à reconnaître qu'une réduction de la complexité entraîne une meilleure compréhensibilité. Dans notre travail nous nous sommes intéressés à plusieurs types d'algorithmes que nous avons évalués dans le chapitre sur le traitement des règles dans les graphes. Les différentes stratégies de simplifications sont réunies dans le sous-menu "Simplification" visible dans la figure 11.9.

Construction interactive des graphes

L'élaboration automatique des graphes est un challenge très important, elle constitue un champ de recherche très fertile en Intelligence Artificielle. Mais il est des problèmes où la connaissance du domaine est primordiale, leur intégration dans la construction du classifieur permet de trouver parmi l'ensemble des solutions performantes celles qui sont les meilleures du point de vue de l'expert : cela peut se traduire par un graphe qui explique mieux le phénomène étudié i.e correspondant à un scénario que l'expert peut comprendre et justifier; cela peut également correspondre à un choix de variables discriminantes moins coûteuses ou plus faciles d'accès.

Afin d'offrir à l'expert un outil suffisamment performant, il nous a semblé important de lui proposer un ensemble d'instruments d'aide à la décision. Cette idée n'est pas nouvelle, rappelons que l'acronyme SIPINA [Zighed, 1985], méthode originelle à la base de nos développements, veut dire "Système Interactif Pour l'Induction Non-Arborescente". Le fait de travailler sous une interface en mode graphique, le bureau Windows® , permet de matérialiser ces instruments à travers des séries de graphiques, en sus des indicateurs numériques, qui résument de manière synthétique les différentes opportunités au niveau du choix de la variable de segmentation.

Au sein de la plate-forme SIPINA_W©, nous avons multiplié ces instruments. Il faut avouer que cela fût, entre autres[65], à l'origine du succès de sa diffusion sur Internet en donnant aux néophytes un logiciel avec une interface agréable, presque ludique. Avec le problème du cancer du sein (Breast Cancer Wisconsin [Murphy et Aha, 1995]), nous illustrons dans les graphiques 11.10 (resp. 11.11) les segmentations concurrentes (resp. les distributions conditionnellement aux classes de la variable à prédire et les indicateurs statistiques afférents) sur un sommet du graphe.

L'interactivité se manifeste également dans la possibilité qui est offerte aux utilisateurs de

65. Nous espérons que ce n'en est pas l'unique raison quand même

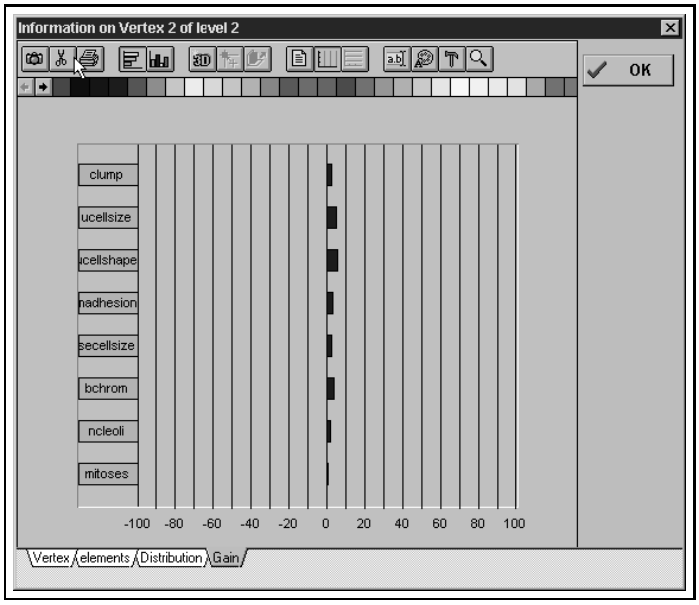

FIG. 11.10 – *"Ucellshape" est la meilleure variable en segmentation, mais on se rend compte que la variable "UcellSize" aurait tout aussi bien induit une partition quasiment de même qualité (au regard du gain d'incertitude)*

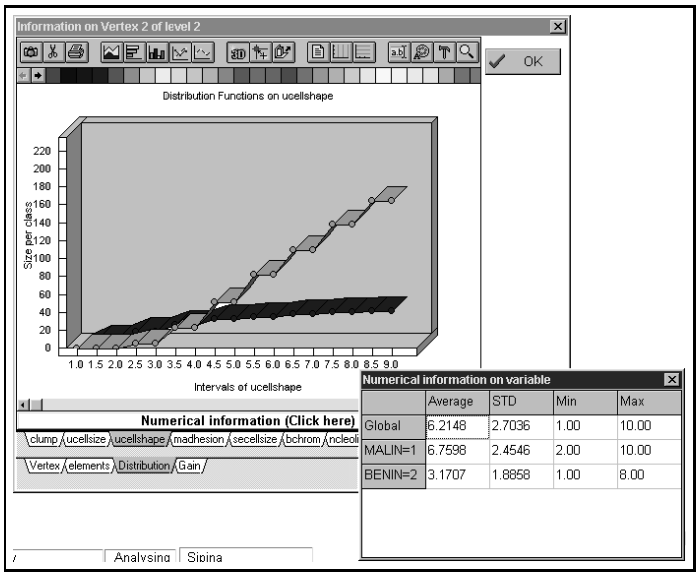

FIG. 11.11 – *Fonction de répartition de l'attribut "UcellShape" conditionnellement aux classes (Malin, Bénin)*

choisir l'opération qui leur convient (fusion, segmentation). De plus, ils peuvent à tout moment restructurer le graphe par un mécanisme de retour en arrière où il est possible de transformer un sommet quelconque en une feuille par un clic de souris.

A moins d'un courage et d'une patience hors pair, on voit mal l'application d'une procédure de cross-validation ou de bootstrap dans ce contexte. La validation empirique des modèles d'experts passe nécessairement par un second fichier dit de validation. A cet effet, il est possible de juger la pertinence du modèle, et même individuellement des règles, en l'appliquant sur ce second fichier. Des indicateurs de qualité de la prédiction, notamment une estimation du taux de bon classement en généralisation, sont facilement accessibles.

11.3.4 Visualisation et mise en forme des résultats

Un des objectifs majeurs de l'apprentissage est l'explication. Ceci éclaire en grande partie le succès des méthodes d'induction de règles face à des classifieurs "boîte noire" plus puissants telles que les réseaux de neurones ou les méthodes des plus proches voisins. Par rapport aux méthodes symboliques classiques issus de l'algorithme AQ [Michalski, 1969], les graphes d'induction possèdent l'incommensurable avantage de pouvoir s'exprimer à travers deux systèmes de représentation : la première, très visuelle, est le graphe de décision où l'on peut suivre le processus de prise de décision en traçant les questions successives menant à la conclusion; la seconde, la base de règles, est plus fonctionnelle tout en restant quand même très compréhensible à l'être humain tant que le nombre de règles est raisonnable.

Au tout début de notre implémentation de la plate-forme, il existait peu de logiciels représentant directement le graphe construit. Seul le système à base de connaissance associé était édité, les interfaces graphiques, notamment sur UNIX, étaient peu répandues et difficiles à programmer. Il en est tout autrement à l'heure actuelle. Les produits représentant le classifieur sous la forme de graphe de décision, avec les distributions de classes associées aux différents sommets, sont nombreux au point que certains commerciaux indélicats affirment que le but premier des graphes d'induction est la structuration des données, que l'on peut visualiser directement. Bien entendu, nous ne pouvons souscrire à de telles allégations. Cependant il est évident que la compréhension du graphe requiert un très faible niveau de compétence de la part d'un non informaticien, il contribue grandement à la popularité de l'outil.

Dans la plate-forme SIPINA_W©, ces deux modes d'expressions du classifieur sont disponibles. Le graphe (figure 11.12), tout comme la base de règles associée (figure 11.13), peut être modifié manuellement par l'utilisateur. Nous avons produit un effort important en faveur de la lisibilité des résultats. Nous avons été bien aidé, il est vrai, par l'environnement de programmation Delphi de Borland.

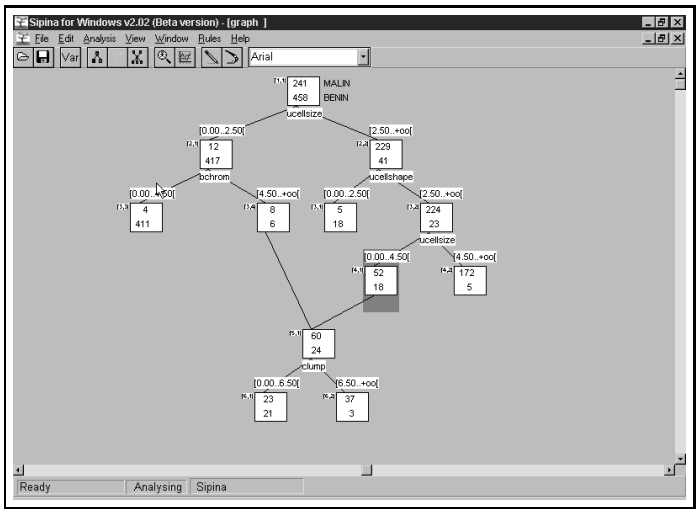

FIG. 11.12 – *Un graphe d'induction construit dans le logiciel Sipina-W(c) sur le fichier des Cancer du Sein*

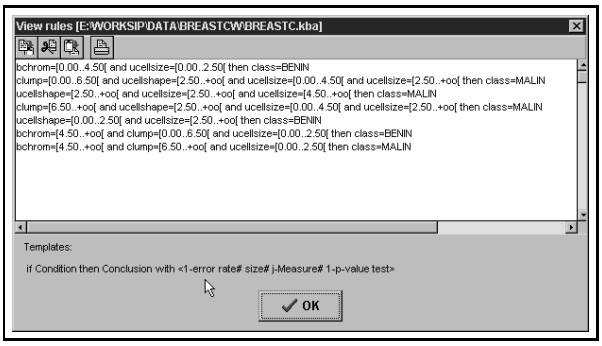

FIG. 11.13 – *Une base de règle construit dans le logiciel Sipina-W(c) sur le fichier des cancer du sein*

11.3.5 Epuration et validation des connaissances

L'apprentissage est éminemment empirique. A partir de régularités décelées dans un échantillon extrait selon une procédure définie par l'expert, nous définissons un classifieur (une partition) sur la population totale. Certains auteurs [Schaffer, 1994] [Wolpert, 1994] [Wolpert, 1996] affirment haut et fort que la connaissance seule de l'échantillon ne permet en aucune manière d'extrapoler sur la population totale, il y a toujours derrière une démarche d'apprentissage plusieurs a priori. Nous ne contestons pas cette affirmation, dans les graphes on suppose que les concepts sont des hyper-rectangles. De l'adéquation de cet a priori avec la réalité dépend alors les performances en généralisation du classifieur. Comment peut-on en juger ?

Nous y avons répondu longuement dans le second chapitre de cette thèse. Nous disposons d'un arsenal d'indicateurs qui nous permet d'évaluer le comportement du classifieur en généralisation, un test statistique permet de vérifier sa pertinence face aux modèles de référence que constituent l'affectation aléatoire de la conclusion au prorata de la distribution des classes dans l'échantillon d'apprentissage, ou encore l'affectation systématique à la classe la plus fréquente.

Le passage aux règles introduit une dimension supplémentaire à la validation du classifieur. Maintenant, nous pouvons évaluer une à unes ses composantes de manière à exclure les règles les moins intéressantes, celles qui dégradent les performances en généralisation. Nous pouvons une fois de plus adopter une démarche empirique en appliquant le classifieur sur un échantillon test, mais nous avons également la possibilité d'utiliser des indicateurs calculés sur l'échantillon d'apprentissage : le taux de bon classement, le nombre d'individus couverts, la j-mesure de [Goodman et Smyth, 1988] et l'intensité d'implication de [Lerman *et al.*, 1981]. Cette dernière notamment peut être utilisée pour valider ou invalider une règle comme nous avons pu le voir plus haut.

11.3.6 Fusion de connaissances

Travailler sur un système de représentation aussi universel que les bases de règles nous autorise une opération supplémentaire : la fusion de connaissances. Les graphes constituent une manière particulière d'extraire des informations des données, ils privilégient certaines formes d'exploration qui peut-être favorisent certaines formes de concepts. Si l'on utilise l'algorithme initial ID3 [Quinlan, 1979], nous avantagerons les prédicteurs introduisant une partition très fragmentaire de l'espace de représentation. En utilisant d'autres méthodes d'induction de règles, il est envisageable de trouver d'autres classifieurs, qui peuvent être au moins aussi intéressants. Nous pensons entre autres à une méthode d'acquisition des connaissances très répandue qu'est l'interview d'expert, ce dernier se soucie peu d'optimisation d'indicateurs numériques, il essaie avant tout de traduire son savoir. Rajouter de telles règles dans la base ne peut qu'améliorer ses performances. Les opérations d'adjonction manuelle et de fusion de bases de règles sont disponibles via le menu

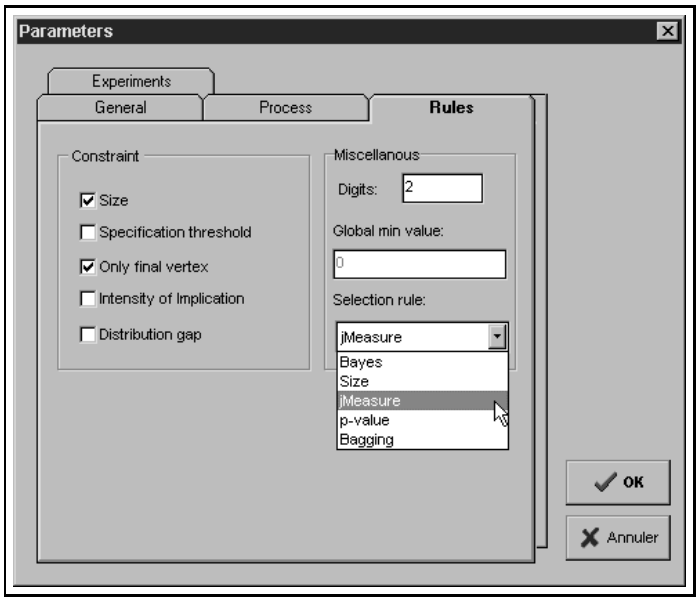

FIG. 11.14 – *Sélection des règles en généralisation*

de manipulation des règles (figure 11.13).

11.3.7 Stratégies de décision et généralisation

Un graphe d'induction induit une partition non recouvrante de l'espace de représentation. Cependant, avec les opérations d'adjonction et/ou de fusions de bases de connaissances, il arrive souvent qu'en généralisation (i.e l'application du classifieur dans la population mère) plusieurs règles peuvent être déclenchées lors du classement d'un individu, menant à des conclusions contradictoires. Il nous faut alors définir des stratégies de décision.

A priori, la méthode la plus simple consiste à choisir la règle qui se trompe le moins dans la population mère. Cette information étant indisponible, nous pouvons choisir l'un des indicateurs de qualité des règles ci-dessus, sous couvert qu'il soit corrélé avec le taux de succès en généralisation dans le problème étudié. Le choix de cet indicateur est accessible via la boîte de dialogue de spécification des règles (figure 11.14).

A la lumière des travaux de [Breiman, 1996a] sur l'agrégation des classifieurs, nous pouvons également adopter une autre démarche en faisant voter les règles. Cela bien sûr pose le problème de la pondération : Faut-il voter à la majorité absolue ? Faut-il accorder aux règles un crédit proportionnel aux valeurs des indicateurs de qualité associés ? Nous ne disposons pas de réponse tranchée à l'heure actuelle, nous constatons empiriquement en tous les cas que la stratégie de vote est nettement supérieure à la sélection de la règle la plus crédible (au vu des indicateurs tels que la j-Mesure ou l'intensité d'implication) dans les bases de connaissances construites par agrégation de prédicteurs issus de bootstrap.

11.4 SIPINA_W©, outil de comparaison des stratégies d'apprentissage

Elaborer un outil pour l'ECD était effectivement un des objectifs de l'élaboration du logiciel SIPINA_W©, la seconde était de construire une plate-forme de mise en oeuvre et d'évaluation de nos travaux de recherche, notamment pour la présente thèse, celle de [Sebban, 1996] ou encore de [Rabaseda, 1996]. Il y eut de nombreuses versions jamais diffusées qui furent le théâtre d'espoirs déçus : soit parce que les idées n'étaient pas viables, soit parce que tout simplement elles n'apportaient pas d'amélioration en performances. Néanmoins, telle qu'elle est constituée, la base permet une évaluation de plusieurs stratégies d'apprentissage. Si l'on recense toutes les possibilités qu'offre la version actuelle, on se rend compte que le nombre de choix possibles est très grand, en effet nous pouvons combiner :

- les méthodes d'apprentissage,
- les règles d'arrêt,
- les méthodes de discrétisation, locales ou globales,
- les contraintes d'extraction de règles.

Les comparaisons peuvent se faire par domaine d'application en chargeant classiquement un fichier afin de comparer deux ou plusieurs techniques d'apprentissage. La comparaison porte sur des résultats issus de validation croisée ou par répétition du couple apprentissage-validation. Pour les évaluations à grande échelle, nous avons imaginé une fiche de commande qui permet de spécifier différents traitements sur plusieurs fichiers successifs, les résultats sont consignés dans un fichier texte désigné par l'utilisateur (figure 11.15). Nous avons utilisé cette procédure pour toutes les comparaisons sur données réelles ou artificielles de cette thèse.

11.5 Conclusion

En tant que féru de programmation, SIPINA_W© a été un des projets les plus enthousiasmants auquel j'ai eu à participer ces dernières années. Au-delà de l'expérience acquise dans la gestion d'un programme d'une telle importance (au moins en taille de code programme), la diffusion du logiciel sur Internet m'a également permis de prendre contact avec de nombreux chercheurs à travers le monde. Si l'on se fie uniquement aux connexions sur notre serveur, il y a près d'un millier de personnes qui ont téléchargé au moins une fois la plate-forme. Ce chiffre est sans aucun doute une borne inférieure puisqu'il ne prend pas en compte les individus qui ont téléchargé à partir des sites miroirs SIMTEL qui étaient notre deuxième mode de distribution.

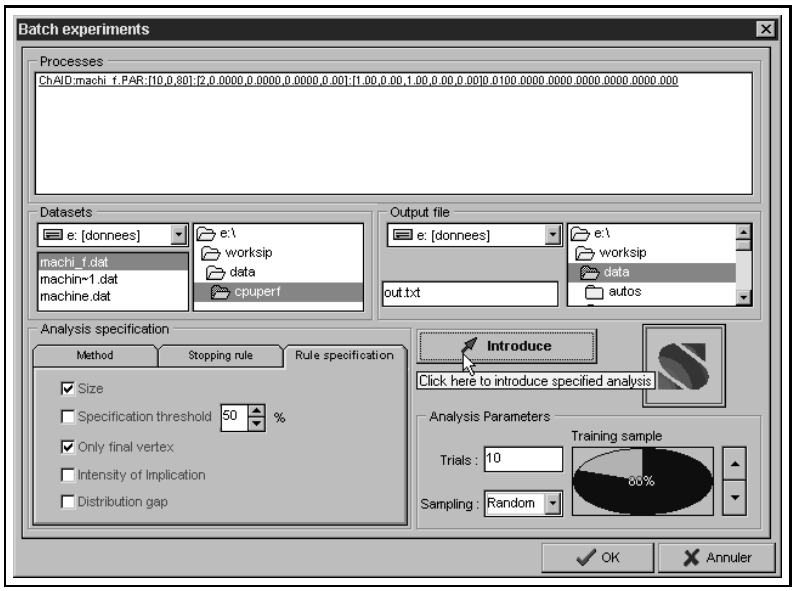

FIG. 11.15 – *Fiche de commande pour le traitement par lots dans Sipina-W(c)*

Ne nous leurrons cependant pas, SIPINA_W© est loin d'être parfait. Certes, il a été exploité avec succès dans différentes études; nous l'avons également mis à contribution dans toutes nos analyses empiriques. Il reste que la base actuelle est irrémédiablement 16 bits, aucune connexion avec les serveurs de bases de données n'est faite. Le passage à une version 32 bits est une occasion unique d'introduire un nouveau saut qualitatif dans l'architecture de la plate-forme :

- une exploitation plus pointue du multi-tâche avec une programmation par threads qui permettra de tirer profit des machines multi-processeurs tournant sous Windows NT®. Cet aspect est d'autant plus intéressant que si l'implémentation des arbres de décision classiques ne pose pas problème, celle des graphes d'induction entraînera vraisemblablement de nombreuses études autour de la synchronisation des processus avec l'opération de fusion;

- l'introduction des modes enrichis de description des données telles que les descripteurs flous [Ramdani, 1994] ou symboliques [Diday, 1995];

- une connexion directe avec les serveurs de bases de données de manière à ce que les produits de requêtes SQL deviennent automatiquement des données en entrée pour l'apprentissage. Apparemment, avec les outils de développement récents, la définition d'architecture client-serveur est sérieusement envisageable avec un coût relativement faible.

Cette liste n'est évidemment pas exhaustive, elle montre en tous les cas qu'il nous reste beaucoup de travail et que la version actuelle est largement perfectible.

Chapitre 12

Les travaux menés à l'aide de la plate-forme SIPINA_W©

12.1 Introduction

Construire un logiciel de Data Mining est une expérience très enrichissante, mais il est évident que c'est à l'usage que l'on est à même de juger de l'efficacité et de la pertinence de notre travail. Fort heureusement, avec l'outil Internet, cela fût aisé. L'acceptation de la référence par les administrateurs de la page WEB "officielle" des *Knowledge Discovery in Databases*[66] a grandement contribué à la diffusion mondiale de notre plate-forme. Les contacts que nous avons établis nous ont permis en grande partie de corriger les défauts de jeunesse du logiciel, ils nous ont également guidés quant à son évolution en terme de fonctionnalités et de capacités de traitement.

La première question à laquelle nous avons voulu répondre est : quels sont les utilisateurs du logiciel? L'expérience nous a montré qu'il est avant tout mis en oeuvre dans une activité de recherche, surtout dans une phase applicative dans des domaines tels que la médecine, la biologie animale... Le recensement des personnes qui ont été personnellement en contact avec nous-même nous permet de dégager quatre principaux contextes d'utilisation :

1. comparaison avec d'autres méthodes dans l'absolu : des chercheurs veulent étalonner leurs méthodes face à la référence éprouvée que constitue l'induction par graphes [Piasta et Lenarcik, 1997];

2. comparaison avec d'autres méthodes dans un domaine précis : sur un problème donné, la culture ambiante (ou encore la tradition) fait que les principales études menées utilisent certaines méthodes d'exploration. L'introduction des graphes constitue une innovation qui permet d'une part de les étalonner en terme de performances en généralisation, d'autre part de vérifier si d'autres formes de connaissances (ici, des règles de production) ne sont pas plus

66. http://www.kddnuggets.com

appropriées pour traduire les concepts. Un exemple précis nous a été par exemple fourni par Mlle Katharina Staërk de l'Université Massey en Nouvelle Zélande : les études menées en épidémiologie animale sont principalement fondées sur des méthodes de scoring, assez efficaces certes mais très peu interprétables. L'introduction des graphes, très peu connus dans ce milieu apporte effectivement le double gain que nous avons cité précédemment. Dans un contexte similaire, [Krall *et al.*, 1996] ont confirmé leurs résultats issus de méthodes multivariées à l'aide d'arbres de décision dans l'étude de l'efficacité des mécanismes de ventilation non-invasive chez des patients hospitalisés.

3. étude exploratoire dans un secteur d'activité précis : on constate que les acteurs les plus dynamiques, en tous les cas qui ont maintenu un contact soutenu avec nous-même, travaillent dans des domaines médicaux ou vétérinaires. [Herr *et al.*, 1997] par exemple a étudié la reconnaissance des chauve-souris à l'aide de leurs cris et a obtenu des taux de succès proches de 95% en utilisant notre implémentation de C4.5 [Quinlan, 1993a]; Mr. Ricardo Rodriguez Iglesias du Département d'Ecologie de l'Université du Texas à Austin a étudié les risques d'érosion en Ethiopie sur la base d'images satellites; Mr. Vincent Danel a exploré les implications autour des comas toxiques [Danel, 1997].

4. support en enseignement : au sein de l'Université Lumière Lyon, le Pr. Zighed et moi-même l'utilisons dans nos enseignements en DESS Statistique et Informatique Socio-Economique et DESS Ingénierie Informatique de la Décision et de l'Evaluation Economique. A l'Université Joseph Fourier de Grenoble, le Pr. Vincent Rial s'en sert dans le DEA Génie Biologique et Médical.

Dans ce chapitre, nous avons décidé de détailler deux études qui ont été menées au sein de notre laboratoire. Nous avons fait ce choix pour deux raisons : ayant pris part activement à ces travaux, nous les connaissons très bien; ils ont été le terrain d'expérimentation de nos développements théoriques. En effet, ne nous voilons pas la face, le succès de notre logiciel parmi les chercheurs repose en grande partie sur la disponibilité de méthodes d'inductions (ID3 [Quinlan, 1986b], CART [Breiman *et al.*, 1984]) autrement plus réputées que nos solutions techniques.

12.2 La reconnaissance des odeurs

Dans son travail de thèse, [Sebban, 1996] s'est beaucoup intéressé à la reconnaissance des odeurs. C'est un problème complexe car il y a une grande part de subjectivité dans notre mécanisme de perception. Le parfum du jasmin par exemple évoque des sentiments plus ou moins forts selon les cultures : culinaires pour les uns, plus poétiques pour les autres. Partant du principe que

la moyenne des subjectivités permet d'atteindre une certaine objectivité, l'auteur a mis au point une expérimentation sur 150 sujets humains destinée à l'alimentation d'une base de données de descripteurs sémantiques liés à une trentaine d'odeurs.

L'idée de base est d'effectuer un traitement à deux niveaux pour "mieux" apprendre :

- le premier consiste à construire une base de connaissance constituée de règles de production issues de la base de données des descripteurs sémantiques où l'on fait tourner la variable à prédire i.e pour chaque descripteur, on essaie de trouver un système de causalité à partir des autres descripteurs pris comme attributs prédictifs. Au regard de l'important nombre de règles que l'on peut dégager ainsi, 42 descripteurs ont été utilisés (16 issus de l'expérience sus-citée, 26 d'une expérimentation menée dans [Dravnieks, 1976]), il a été nécessaire de procéder à un élagage drastique qui a été réalisé à l'aide de l'intensité d'implication de [Lerman *et al.*, 1981], ainsi nous avons pu déceler les causalités fortes. Ces règles servent alors à alimenter un système expert (le système *SelfMind*[67]) travaillant en logique multi-valuée qui a été développé au sein de notre laboratoire par Fabrice Muhlenbach.
- le second traitement est numérique, il consiste à projeter les individus au sein d'un espace à 5 dimensions défini par des variables physico-chimiques issus des travaux de [Patte *et al.*, 1982].

La prédiction de l'odeur d'une nouvelle observation revient alors dans un premier temps à le projeter dans l'espace ci-dessus; dans un second temps détecter son voisin le plus proche (au sens d'un voisinage particulier) dont on utilise dans un troisième temps les propriétés sémantiques dans l'inférence à partir du système expert ci-dessus. Dès lors, "mieux apprendre" pour nous prend un sens assez particulier : les descripteurs permettent de déclencher des règles qui mettent en évidence la classe à prédire mais également les autres propriétés sémantiques associées à l'observation.

Au delà de la contribution des travaux de [Sebban, 1996] dans l'étude de la modalité olfactive qui a donné lieu à de nombreuses publications, nous retiendrons pour notre part l'extraordinaire champ d'investigation qu'a représenté cette étude pour l'évaluation et la validation des règles. La causalité entre prémisse et conclusion que nous avons quantifiée à l'aide de l'intensité d'implication nous a permis de réduire les bases de règles dont le gigantisme menaçait les performances du système de décision, elle a également permis la résolution des conflits dans les cas de conclusions contradictoires en définissant une hiérarchie entre les règles.

12.3 La caractérisation de la marche

En collaboration avec l'Hôpital Henry Gabrielle à Lyon, [Rabaseda, 1996] a travaillé sur l'analyse automatique de la marche à des fins "de diagnostic, d'évaluation thérapeutique et d'expertises

67. Système Expert en Logique Floue et Multi-valuée pour l'Imitation Naturelle des Décisions

médico-légales des incapacités". De manière schématique, il s'agissait en fait de reconnaître différents types de boîteries en fonction de paramètres de déplacement en trois dimensions mesurés sur des patients.

Le sujet s'inscrivait dans la problématique de l'extraction automatique de connaissances à partir de données, les principales phases y étaient présentes : recueil des données, préparation pour l'exploitation, traitement à travers plusieurs algorithmes d'induction, évaluation et interprétation des connaissances acquises. En fait, cette étude de la marche a été le cadre en premier lieu de tous les développements réalisés en matière logicielle intégrés dans SIPINA_W©, notamment le module d'évaluation empirique des connaissances (validation croisée, apprentissage répété) et la comparaison de la méthode SIPINA [Zighed *et al.*, 1992] avec d'autres algorithmes d'induction de règles [Breiman *et al.*, 1984] [Quinlan, 1993a].

Outre ces méthodes, d'autres stratégies d'apprentissage ont été introduites aux fins de comparaisons : la régression logistique [Giraud, 1993], l'analyse discriminante [Fisher, 1936], deux extensions de l'algorithme de l'étoile [Sebag, 1995] [Venturini, 1994], un algorithme symbolique d'exploration sélective du treillis de Gallois [Ganascia, 1987]. Les critères de comparaison ont été les performances en généralisation et le nombre de règles lorsqu'il est mesurable.

Le protocole d'expérimentation qui a été mis en place est assez particulier, l'objectif était plus de discerner les caractères discriminants entre les différents types de boîteries, et non pas la reconnaissance en elle-même. Ainsi, on a mesuré les paramètres de démarche de 20 individus sains à qui par la suite on a posé des prothèses spéciales simulant plusieurs niveaux de boîterie de plus en plus contraignantes. Il est évident que les performances en généralisation ne constituent pas une fin en soi ici, l'intérêt de cette étude réside avant tout dans l'analyse des paramètres de différenciation entre ces niveaux. Au final, l'auteur constate que seuls les deux derniers stades de la boîterie simulée, correspondant à des fortes gênes, sont véritablement discernables de la démarche normale.

Outre sa contribution à l'étude de la marche au sein du laboratoire de recherche de l'Hôpital Henry Gabrielle, le principal intérêt de cette collaboration est d'avoir mis l'accent sur un aspect trop souvent sous-estimé de l'apprentissage en Reconnaissance de Formes : l'analyse des résultats pour l'explication. Dans cette optique, l'aspect interactif et visuel du logiciel a été largement mis à contribution, dans la lecture des graphes comme dans la lecture des règles de production.

12.4 Conclusion

La plupart des expériences citées dans ce chapitre ont été pour nous autant d'occasions d'échanges d'idées qui nous ont guidés dans les spécifications de ce que pourrait être une plate-forme complète d'extraction de connaissances à partir de données. Dans ce cadre, nous tenons à remercier toutes ces personnes pour leur patience face aux "quelques" bogues qui ont émaillé les

versions successives, et plus particulièrement Mr. Hani Iskandar dont les suggestions nous ont été bien utiles dans bien des cas.

Enfin, si la plupart de nos contacts ont été des étudiants en thèse, plus disponibles, ces échanges nous ont également amené à rencontrer des personnes assez originales qui ont exploité SIPINA_W© dans un cadre où on ne l'attendait pas vraiment. Ainsi M. Renato Podesti, docteur en électronique et analyste programmeur indépendant, a décider d'étudier attentivement les résultats des courses de chevaux à Rome. Dans un contexte de données très bruitées, il a constaté que la version de CART avec une règle de segmentation TWOING donnait les meilleurs résultats en prédiction. Au final, il affirme obtenir une amélioration de 50 à 60% par rapport à ses gains habituels, ce qui nous donne un petit espoir pour notre avenir si d'aventure l'après thèse s'avèrait difficile.

Chapitre 13

Conclusion

Les graphes d'induction ont été, et seront encore pour de nombreuses années, le théâtre de maints travaux. Les points abordés dans cette thèse montrent combien la construction "classique" des graphes, en une seule passe sur un fichier d'apprentissage, est maintenant arrivée à un niveau de sophistication assez élevé. Qu'en est-il de l'avenir de la recherche dans les graphes, peut-on encore en améliorer les qualités ? Dans quel sens ? Quel avenir ont-ils du point de vue de la vulgarisation et de son applicabilité sur les problèmes réels ? Nous essaierons de répondre à ces questions dans ce dernier chapitre, mais auparavant essayons de faire le bilan de cette thèse.

13.1 Constats et conclusions

Lors du choix du plan de présentation de notre travail, nous étions conscients de prendre un risque fâcheux, celui de noyer au milieu des travaux des autres auteurs nos propres contributions. Pourtant nous l'avons pris parce qu'il nous semblait nécessaire d'apporter une vision fédératrice des travaux sur les graphes d'induction tant les propositions ont été nombreuses ces dernières années : certaines structurées, reposant sur des arguments théoriques forts; d'autres plus folkloriques, justifiées surtout par le bon sens et une amélioration, parfois fallacieuse, des performances sur des bases exemples. En proposant un cadre commun à ces approches, ou du moins en essayant de situer le contexte de leur développement, nous espérons pouvoir offrir un document de travail de base afin que nous-même, et pourquoi pas d'autres, puissions définir en connaissance de cause les axes de recherche futurs. En ce sens, nous pensons que la principale contribution de cette thèse est d'avoir fait un état de l'art approfondi en délimitant au mieux le champ que nous couvrons. Une vie entière ne serait pas suffisante pour faire un survol exhaustif, on peut se demander même si cela est nécessaire. Fidèles aux principes de l'apprentissage, nous avons surtout essayé de mettre en évidence les principaux invariants.

Afin de souligner les conclusions auxquelles nous sommes parvenus, nous reprenons les thèmes

développés dans cette thèse en nous inspirant du plan de présentation :

- évaluation empirique des modèles : c'est un domaine fondamental en apprentissage, notre objectif est de construire un prédicteur qui soit le reflet des mécanismes de causalité dans la population totale. A partir d'un échantillon d'apprentissage, il est difficile de s'en assurer de manière certaine, on doit se tourner alors vers des méthodes statistiques d'estimation qui ne donnent pas des solutions "vraies", mais proposent des réponses avec une fiabilité mesurable. Néanmoins, pour que les estimations et comparaisons soient valables, il est nécessaire de s'entourer de moultes précautions, au niveau de la définition de l'expérimentation ainsi qu'au niveau des conclusions. Un des fléaux les plus pernicieux que nous avons dénoncé est le postulat de supériorité d'un algorithme sur les autres, entièrement fondé sur des résultats expérimentaux établis à partir de bases réelles distribuées sur des serveurs WEB. Le rôle essentiel de ces fichiers n'est pas en cause, c'est plutôt l'utilisation que l'on en fait qui pose parfois problème. A notre sens, il est illusoire de se battre sur des taux de succès en validation si l'on n'arrive pas à cerner les caractéristiques du domaine sur lequel on travaille. Quant à la généralisation intempestive sur la foi d'une supériorité plus fréquente sur les bases utilisées, il est clair que l'hypothèse qui la sous-tend est rarement vérifiée, à savoir que les fichiers sont représentatifs des problèmes réels que l'on peut rencontrer en apprentissage. Même si l'on adopte la règle de l'unanimité (A est meilleur que B si et seulement si il est meilleur sur tous les fichiers exemples), la conclusion peut être entachée d'erreur. Pour s'assurer de la représentativité, la multiplication du nombre de bases de test n'est pas non plus une solution. Nous pensons que la principale utilisation de ces fichiers est illustrative, ils servent à montrer la viabilité du nouveau paradigme proposé, et surtout à caractériser leurs conditions d'applicabilité, ainsi qu'à expliquer leur meilleur comportement dans certains cas. De toute manière, à la lumière de la plupart des papiers sortis ces dernières années, les progrès en taux de reconnaissance, si l'on s'en tient aux algorithmes "classiques"[68] sont faibles sur ce type de données depuis les références ID3 [Quinlan, 1986b] et CART [Breiman *et al.*, 1984].

- mesures d'évaluation des segmentations : clairement, il est apparu dans ce chapitre que l'enjeu n'est plus seulement la recherche des prédicteurs les plus précis, mais plutôt de donner certaines qualités aux graphes produits. L'une d'elle est sans conteste la capacité à produire des classifieurs de petite taille. On sait depuis les travaux de [Holte, 1993] que sur les données réelles, les partitions de faible complexité sont une bonne approximation du concept à apprendre, la mesure d'évaluation des arbres doit donc assurer une exploration de l'espace des solutions performante tout en évitant un éparpillement excessif des données. Notre travail a consisté à classer les mesures selon la nature de l'information qu'elles

68. coupures "dures" et apprentissage unique

utilisent (distance, gain informationnel...), puis à les positionner par rapport à une série de propriétés de "bonne tenue" définies et justifiées dans les travaux de [Zighed *et al.*, 1992]. Il est intéressant de noter que certaines d'entre elles, les propriétés de sensibilité à la taille de l'effectif et de fusion, précisent de manière analytique des qualités que l'on voulait affecter à ces mesures pour que l'on puisse évaluer non plus la qualité d'une segmentation mais plutôt la qualité d'une partition.

- détermination de la bonne taille du graphe : c'est le deuxième point clé de l'induction par graphes. Qu'importent finalement les raisons qui justifient la recherche de la bonne taille, on sait tout simplement qu'elle influe sur les performances du classifieur. Pendant longtemps, les solutions ont reposé sur des heuristiques d'estimation du taux d'erreur en généralisation. Le paradigme bayésien a apporté une rigueur qui a permis de basculer le problème d'induction vers un problème d'optimisation que l'on situe mieux et qui a permis d'ouvrir de nouvelles voies de recherche. Certains chercheurs en ont fait leur religion, il apparaît maintenant qu'il ne s'agit que d'une solution parmi d'autres qui garantit l'arbitrage entre la faible complexité du graphe et sa précision sur le fichier d'apprentissage. Si l'une ou l'autre pèse trop, les performances seront dégradées. On remarquera que l'utilisation des mesures d'évaluation des partitions sont essentielles dans ce saut conceptuel, nous avons pu montrer que l'induction par graphes pouvait être rapprochée du schéma général de régression d'optimisation de la corrélation entre les attributs prédictifs et la variable à prédire [Rakotomalala et Zighed, 1997].

- extraction des règles : assimiler les graphes d'induction aux méthodes symboliques d'induction de règles serait une erreur regrettable. Tout d'abord parce qu'ils utilisent leur propre système de représentation de connaissances, utilisable directement dans l'explication et dans la prédiction; ensuite parce que le passage aux règles n'est pas une opération triviale, elle peut être l'occasion de nombreuses améliorations du classifieur. L'utilisation d'indicateurs de qualité des règles, proches d'ailleurs dans leur principe des indicateurs de qualité des partitions, permet d'hiérarchiser les règles mais surtout de les valider au sens d'un test statistique. Dès lors, nous pouvons adopter des stratégies plus évoluées d'extraction de règles dans les graphes en sélectionnant les meilleurs sommets [Rakotomalala et Zighed, 1996]. De fait, la base de règles associée au classifieur est homogène au sens du test et nous pouvons appliquer des algorithmes symboliques de simplification. Ce deuxième volet, la simplification, ouvre des perspectives assez intéressantes. Le changement de système de représentation permet d'introduire de nouveaux opérateurs qui s'avèrent particulièrement efficaces en réduisant de manière drastique la complexité du classifieur. Enfin, si le schéma classique de l'apprentissage supervisé cherche avant tout à désigner une classe conclusion, nous pouvons également envisager l'exclusion d'une classe

dans un sous-ensemble circonscrit par une prémisse. Dans les problèmes à plus de deux classes, l'information ainsi introduite peut s'avérer décisive dans le cadre de l'insertion de la règle dans un système à base de connaissances où les conclusions peuvent devenir des entrées dans le processus d'expertise.

- enrichissement du pouvoir de représentation du modèle : partant du système de représentation originel sous forme d'arbre de décision, le seul développement véritablement marquant en vingt années de recherche est la généralisation aux graphes qui permet de traduire plus simplement des concepts plus complexes. Contrairement à la construction inductive où l'on essaie itérativement de donner un meilleur pouvoir de discrimination aux attributs prédictifs, ce saut qualitatif traduit une évolution naturelle si l'on se réfère à la théorie des graphes. Il est vrai que si l'on veut des classifieurs plus riches, il en existe plusieurs dans la littérature comme les réseaux de neurones ou encore les plus proches voisins. L'avantage ici est que l'on ne compromet pas la lisibilité du modèle et le passage au règles est tout aussi naturel. Il semble cependant que dans les cas réels, cet avantage théorique ne se manifeste vraiment que dans certains cas particuliers : lorsque le concept "matche" bien avec la structure du graphe, lorsque l'effectif d'apprentissage est de taille relativement réduite.

- enrichissement du pouvoir de discrimination des variables : l'inférence inductive est le pendant de la démarche précédente, au lieu d'améliorer le système de représentation on cherche à améliorer le pouvoir de discrimination des variables en utilisant des combinaisons de deux ou plusieurs attributs. L'objectif ici est toujours de réduire la complexité du classifieur en évitant la réplication des sous-arbres et la fragmentation des données. Dans la pratique encore une fois, l'avantage que l'on constate sur des données artificielles ne se manifeste pas sur données réelles. Est-ce que ce fait est imputable à la nature des données utilisées (concepts trop simples, inadéquation des arbres...) ? Il est clair que le système de représentation choisi ne peut être qu'une approximation du véritable concept (si tant est que le véritable concept existe) surtout lorsque les attributs sont pour la plupart réels. Dans ce cas, la transformation des variables est très importante, même si la multiplicité des solutions impose des biais de préférences très contraignants pour l'exploration des solutions. Le véritable dilemme est alors l'alternative : construction des variables intermédiaires pendant l'induction ou construction des variables dans une phase préparatoire et interprétation préalable. Cette dernière solution est inscrite dans le processus ECD et nous penchons volontiers vers cette solution car la combinaison automatique de variables, autres que booléennes, pose toujours le problème de l'interprétabilité du classifieur, qui est un des atouts clés des graphes d'induction.

- discrétisation des attributs continus : c'est un domaine sur lequel nous avons beaucoup travaillé. En matière de discrétisation à l'aide d'une borne "dure", il y a peu de champs que nous n'ayons explorés dans le cadre de la construction des graphes d'induction. Nous avons ainsi pu constater que l'adoption d'une stratégie optimale est peu décisive dans la pratique [Zighed *et al.*, 1997]. Pourtant le sujet est loin d'être clos, loin de là. Nous pensons que ramener la discrétisation à un problème d'estimation statistique est une approche très prometteuse [Wehenkel, 1997]. Tant que l'on travaille sur un échantillon, la généralisation sur la population comporte une part d'incertitude que l'on doit traduire. On peut le faire en termes de biais et de variance de l'estimateur. Dans d'autres approches telles que la théorie des ensembles flous, on l'exprime à travers des fonctions d'appartenance. En tous les cas, malgré les avancées très importantes depuis les algorithmes primitifs de découpage en intervalles ou effectifs égaux, et malgré les allégations de plusieurs auteurs concernant la stérilité de ce domaine de recherche face à une absence d'amélioration des performances par rapport à des méthodes simples, nous pensons que d'importants résultats, ne serait-ce qu'en terme de formalisation, sont encore à prévoir.

- agrégation des classifieurs : c'est le dernier champ de recherche à la mode. A vrai dire, les méthodes développées au fil des années sont générales aux différents types de prédicteurs. Historiquement, en revanche, après un rapport technique resté fameux [Breiman, 1994], les premiers travaux ont été réalisés à l'aide des graphes qui présentent l'énorme avantage d'être facilement programmables, des sources sont même disponibles sur différents sites WEB pour les expérimentations. Si les performances sont incontestablement au rendez-vous, il en est autrement en ce qui concerne la justification théorique où les avis restent partagés. Du reste nous sommes assez sceptiques quand à l'opportunité de la recherche dans cette voie. En effet, si le vote améliore les performances, la grande quantité de graphes générés empêche la lisibilité du modèle. Si l'on perd cette qualité, autant passer aux modèles "boîte noire" réputés plus puissants tels que les réseaux de neurones. D'un autre côté, des tentatives comme celle de [Breiman et Shang, 1996] laissent entrevoir des voies de recherche enthousiasmantes, on peut ainsi se demander si la traduction d'autres classifieurs en graphes d'induction utilisant la même approche ne sont pas des opportunités que l'on doit considérer attentivement.

13.2 Perspectives

En ce qui concerne les graphes d'induction, les enjeux en termes de performances reposent sur deux axes importants : l'augmentation de la précision en généralisation et la réduction de la complexité. Au vu de la plupart des travaux et analyses que nous avons consultés, il apparaît

maintenant que les graphes "classiques", construits à l'aide d'un apprentissage unique sur un échantillon de la population, ont atteint leurs limites. Certes il existe des pistes de recherche assez récentes qui donnent de bons résultats comme la sélection des variables, la réduction des individus,... mais souvent les résultats sont surtout probants sur la réduction de la taille du graphe et non sur la réduction du taux d'erreur en généralisation. A moins d'innovations exceptionnelles, l'augmentation de la précision dans le contexte de l'apprentissage que nous qualifions de classique semble problématique.

Il en est tout autrement en revanche si l'on considère d'autres pistes de recherches que [Dietterich et Kong, 1995b] recensent dans leur article. En partant d'une interprétation du biais et de la variance de l'erreur dans le cas de fonction de perte de type 0/1, ils discutent des améliorations présumées des différents axes de recherche. Partant du constat que le système de représentation est fixe, la réduction du biais qui traduit l'incapacité du classifieur à saisir le concept à apprendre paraît peu probable. De fait, les méthodes plébiscitées sont celles qui réduisent la variance en restituant l'incertitude liée au classement. Hélas, ces stratégies comme l'agrégation des classifieurs ou la production d'arbres flous[69], même si elles donnent de bons résultats, réduisent à néant l'avantage des graphes par rapport aux autres méthodes d'apprentissage, à savoir la lisibilité du processus de décision. Certes, il existe des tentatives louables comme les arbres à options [Kohavi et Kunz, 1997] qui traduisent l'incertitude sus-dite tout en s'approchant des performances des classifieurs agrégés, mais appliquées sur des bases réelles, la complexité de l'arbre devient rédhibitoire.

Malgré tout, nous sommes fermement convaincus que l'avenir des graphes en induction est réel, ne serait-ce que comme cadre d'exploration de nouvelles formulations et de tentatives d'explicitations théoriques. Ce type de résultat compte d'ailleurs parmi les aspects positifs de notre travail et de son positionnement dans notre équipe de recherche : les réflexions autours de l'optimisation en apprentissage et les performances en généralisation que nous avons étudiées dans le cadre de la discrétisation ont pu être étendues à d'autres algorithmes d'induction de règles [DiPalma *et al.*, 1997]; la caractérisation des règles aux fins d'extraction dans les graphes a pu être étendue sur l'algorithme CN2 [Rakotomalala *et al.*, 1997b]; les travaux autour de l'appréciation des mesures d'évaluation des partitions nous ont permis de replacer l'induction par graphes au sein du schéma général de régression [Rakotomalala et Zighed, 1997]; nos premiers essais autour de l'agrégation des classifieurs ont été relayés par des approches plus sophistiquées qui semblent prometteuses [Gavin *et al.*, 1997].

Enfin, si nous nous tournons vers les applications existantes, il est clair que les graphes possèdent un avenir indiscutable. La prolifération des logiciels utilisant ce paradigme est un signe qui ne trompe pas. Dans le cadre de l'ECD notamment, les graphes se positionnent comme la

69. La situation des arbres flous est un peu particulière, sur le graphe on peut concevoir de suivre le processus de décision. La traduction en règle est par contre très problématique.

méthode reine, celle qui associe le mieux explicabilité et précision. De plus, sa rapidité d'exécution et sa robustesse sont d'un excellent niveau face à des algorithmes plus sophistiqués. A terme, on peut espérer que les graphes entreront dans la catégorie des outils d'analyse "grand public", au même titre que les méthodes statistiques classiques telle que la régression simple.

Annexe A

Description des bases de données tests utilisées dans la thèse

Comme on a pu le souligner dans le second chapitre de cette thèse, l'étude des algorithmes sur les bases exemples issues des serveurs de données telles que l'UCI Irvine [Murphy et Aha, 1995] n'est pas un processus trivial. Suite aux travaux de [Mingers, 1989b] qui a basé son analyse sur 4 fichiers de données pour aboutir à des conclusions complètement érronées, il est apparu aux chercheurs en apprentissage qu'un certain nombre de tests était nécessaire pour analyser véritablement le comportement des méthodes. Mais multiplier les applications ne sert pas à grand chose si l'on ne caractérise pas les fichiers utilisés afin de situer l'efficacité ou la déficience des méthodes d'apprentissage. C'est ce à quoi nous allons nous atteler dans cette annexe.

Il existe plusieurs critères pour apprécier et différencier les fichiers de données :

- type de données (artificielles, réalistes, réelles);
- domaines d'étude (médecine, traitement du signal,...);
- forme du concept à apprendre et niveau de bruit;
- type des variables prédictives (continues, qualitatives booléennes et/ou multivaluées, mixtes);
- nombre de classes à apprendre, distribution initiale;
- taille du fichier d'apprentissage.

Certains auteurs pensent que choisir les fichiers de données de manière à effectuer un amalgame raisonnable sur la base de ces propriétés permet de couvrir une vaste classe d'applications [Buntine, 1991]. Cela est indéniable, encore faut-il préciser le positionnement des fichiers par rapport à ces critères.

Fort heureusement, les serveurs proposent des fichiers bien documentés où l'on retrouve l'essentiel de l'information se rapportant aux critères ci-dessus. Nous disposons également des informations sur les performances de diverses stratégies d'apprentissage, ce qui permet la comparaison directe avec nos résultats.

Le choix de fichiers que nous avons réalisé résulte d'une double volonté. D'une part, ne voulant pas être influencé par le domaine d'étude, nous avons procédé à un tirage aléatoire à partir des bases disponibles dans le serveur Irvine; d'autre part, il est évident que certaines données telles que les fichiers des Iris ou encore des Ondes (Wave), constituent des "classiques" indiscutables de la littérature, il était impensable de les exclure ne serait-ce que pour étalonner nos implementations.

Au total nous avons sélectionné quinze bases que nous décrivons succinctement ici :

autos on cherche à identifier le risque associé à des automobiles à partir d'informations sur ses caractéristiques. "Risque" ici est un terme utilisé en assurances et doit être compris relatif au prix i.e par rapport à son prix, est-ce que la voiture implique plus ou moins de remboursements? Les niveaux sont établis par les experts.

breast il s'agit du fichier "Breast Cancer Wisconsin". Très fréquemment cité dans les études[70], il s'agit de prédire l'occurence du cancer du sein chez des femmes à partir d'attributs décrivant l'apparence des cellules. Notons que le fichier complet que nous utilisons (699 individus) proviennent de 8 campagnes d'enquêtes étalées dans le temps. Nous n'avons pas testé l'homogéneité des données mais au vu de la qualité des résultats (97% de succès en moyenne sur toutes les méthodes), on peut penser qu'aucune modification structurelle n'a eu lieu sur la période considérée.

car ce sont des données assez particulières. En effet, la variable à prédire (évaluation des modèles) a été définie par un système expert à partir d'informations techniques sur les automobiles. La question est : peut-on reconstruire ce processus de décision?

cpuperf on veut connaître les performances relatives de processeurs en utilisant leurs caractéristiques techniques telles que la taille mémoire, le nombre de canaux... La variable à prédire est à l'origine continue, elle a été discrétisée en utilisant des bornes fournies par les auteurs. Ces données ont été également le cadre d'autres estimations s'appuyant sur la régression linéaire. Nous avons inclu la variable ainsi estimée parmi les attributs prédictifs fournis par les auteurs, ce qui explique les bonnes performances relevées ici. Dans le cas où nous excluons cette variable, le taux d'erreur en généralisation est proche de 65% en moyenne.

70. O. L. Mangasarian and W. H. Wolberg: "Cancer diagnosis via linear programming", SIAM News, Volume 23, Number 5, September 1990, pp 1 & 18.

credit l'objectif est l'approbation des crédits. Pour des soucis de confidentialité, les noms des attributs ont été modifiés, ce qui empêche toute interprétation. Le fichier est intéressant parce qu'il mélange plusieurs types d'attributs prédictifs (booléens, nominaux, continus).

flags on cherche à reconnaître la région d'origine d'un drapeau national en utilisant des caractéristiques diverses comme la langue nationale, la taille de la population, la religion dominante... Ici également, le mélange des types d'attributs prédictifs est assez intéressant.

hepatitis dans le domaine médical, on essaie de diagnostiquer l'occurence d'une hépatite chez des individus. On remarque dans ce fichier que la distribution initiale des classes est très désequilibrée, on a constaté a posteriori que très peu de méthodes d'induction de graphes ont réussi à éviter ce piège.

ionosphere en traitement du signal, il s'agit de déterminer dans quelles conditions les emissions permettent de reconnaître certaines structures dans l'ionosphere. Tous les attributs sont continus.

iris c'est le fichier incontournable de la reconnaissance de formes du à Fischer (1936). Une classe est linéairement discriminable, les deux qui restent en revanche sont légèrement recouvrantes dans l'espace de représentation. Une variable prédictive suffit pour obtenir de bons résultats.

lung-cancer il y a très peu d'informations sur ce fichier, sa principale particularité est la présence d'attributs prédictifs en nombre supérieur par rapport au nombre d'observations. La rareté des observations d'ailleurs nous pousse à une extrême prudence quant à la validité des résultats enregistrés.

pima toujours dans le domaine de la médecine, chez une population féminine habitant près de Phoenix, en Arizona, on veut diagnostiquer l'apparition de signes de diabetes en utilisant la définition de l'Organisation Mondiale de la Santé.

vote le concept à apprendre est simple, le "meilleur" prédicteur est un arbre trivial à un niveau. On veut reconnaître l'appartenance politique d'un parlementaire américain à travers son comportement de vote sur des problèmes telles que l'accord de crédits au ministère des armées, l'adoption du budget,...

wave éminemment classique, les ondes de Breiman sont des données artificielles dont on connaît les principales caractéristiques, notamment le taux de reconnaissance théorique du prédicteur bayésien. Ce fichier est repris quasiment dans toutes les publications en apprentissage automatique.

Fichier	*Observ.*	*Attributs*	*Attr. Cont.*	Classes	*Min. Précision*
autos	205	25	15	6	0.33
breast	699	10	10	2	0.65
car	1728	6	0	4	0.50
machine	209	8	8	8	0.59
credit	690	15	6	2	0.55
flags	194	28	10	6	0.27
hepatitis	155	19	6	2	0.79
ionosphere	351	34	34	2	0.64
iris	150	4	4	3	0.33
lung-cancer	32	56	0	3	0.40
pima	768	8	8	2	0.65
vote	435	17	0	2	0.61
wave	5000	21	21	3	0.33
wine	178	13	13	3	0.40
zoo	101	8	1	7	0.41

TAB. A.1 – *Description des données tests*

wine on veut reconnaître des vins en provenance d'Italie à partir de leur composition chimique. Tous les attributs prédictifs sont continus. On note qu'il s'agit de la concaténation de trois bases, apparemment l'hétérogénéité des données ne pose pas problème ici puisque certaines méthodes approchent le 100% de reconnaissance.

zoo ce fichier est très mal documenté. On veut reconnaître le type d'un animal (la variable à prédire dont on ne sait pas quelle est la définition exacte) en utilisant les caractéristiques physiologiques. Tous les attributs, mis à part la longueur des pattes, sont booléens.

Nous recapitulons dans le tableau A.1, les informations quantitatives relatives à ces fichiers. Notons qu'afin d'éviter toute forme d'intérférence, nous avons exclu les observations contenant au moins un attribut manquant. Ce n'est certainement pas la meilleure stratégie pour l'apprentissage, elle garantit en revanche une réelle objectivité pour les comparaisons : chaque méthode dispose de la même information pour l'inférence. Les caractéristiques recensées sont : le nombre d'observations, le nombre total d'attributs prédictifs, le nombre d'attributs prédictifs de type continu, le nombre de classes de la variable à prédire, la précision atteinte si l'on choisissait la classe la plus fréquente dans l'échantillon d'apprentissage.

Annexe B

Comparaisons des performances des méthodes implémentées sur les bases tests

Nous avons implémenté, dans le logiciel SIPINA_W©, plusieurs méthodes d'induction par graphes que l'on retrouve dans la littérature. Notre objectif dans cette expérimentation à grande échelle est triple :

- évaluer les stratégies dans des conditions réelles d'utilisation, les heuristiques utilisées reprennent à la lettre les indications des auteurs respectifs.
- comparer leurs performances. Contrairement à la démarche que nous avons adopté dans notre texte où nous comparons des algorithmes qui ne diffèrent que sur l'élément que nous voulons éprouver (mesure, passage aux graphes...), nous confrontons ici des méthodes complètement hétérogènes où le rôle de certaines heuristiques peuvt être masqué (e.g l'élagage peut rattraper la propension au sur-apprentissage d'une mesure).
- étalonner notre implémentation face aux résultats publiés par différents auteurs dans la littérature existante (surtout la série de tableaux de comparaisons publiée par Quinlan qui a pris un malin plaisir à multiplier les résultats empiriques [Quinlan, 1996] [de Merckt et Quinlan, 1996] [Quinlan, 1996]).

Dans notre comparaison, nous avons utilisé une validation croisée stratifiée en 10 parties. Les résultats sont consignés dans le tableau B.1 (resp. B.2) pour le taux d'erreur en généralisation (resp. le nombre moyen de règles générées).

Sur le taux de succès en validation, quelques méthodes semblent significativement se distinguer, mais aucune n'est supérieure aux autres sur l'ensemble des bases. Sur le nombre de règles

Fichier	C4.5	CART_G	CART_T	ChAID	Elisee	ID3	SIPINA	WDT	QR_MDL	PRETree
autos	72.3	65.5	57.3	68.5	68.5	69.3	66.5	66.4	69.7	70.4
breast	94.7	95.1	95.6	95.3	94.9	96.1	95.9	95.9	95.9	98.1
car	87.3	87.3	87.3	87.2	86.8	87.3	86.8	86.8	87.3	88.7
machine	97.1	95.2	95.2	92.0	92.0	94.3	94.3	96.2	97.1	96.1
credit	84.1	62.8	97.6	92.3	98.0	92.3	91.4	92.4	90.5	89.3
flags	50.4	51.4	64.7	55.1	65.9	63.0	44.1	68.5	59.0	60.0
hepatitis	77.5	78.7	79.5	88.4	88.4	80.7	78.3	80.8	82.4	82.0
ionosphere	91.0	88.1	90.0	85.9	85.9	84.2	88.0	91.1	88.7	93.1
iris	96.7	93.3	92.7	96.7	96.7	94.7	94.7	94.7	96.7	97.3
lung-cancer	53.3	50.0	57.3	58.7	58.7	50.0	50.0	58.7	51.3	53.3
pima	72.7	75.6	74.9	75.5	75.5	75.9	80.5	74.0	74.9	71.2
vote	94.8	95.0	95.3	91.1	94.0	94.5	95.0	95.0	94.3	96.8
wave	67.0	65.3	64.3	70.4	73.4	70.0	71.0	70.0	70.7	60.0
wine	87.7	81.6	85.6	88.7	88.7	82.8	86.1	85.6	86.0	91.0
zoo	88.1	87.3	82.8	79.1	79.1	87.0	88.0	81.0	83.7	87.9

TAB. B.1 – *Taux de succès moyen en généralisation sur les méthodes implémentées dans SIPINA-W*

Fichier	C4.5	CART_G	CART_T	ChAID	Elisee	ID3	SIPINA	WDT	QR_MDL	PRETree
autos	29.3	14.8	18.3	16.0	16.0	17.3	7.6	10.7	24.6	31
breast	25.3	7.7	8.6	14.7	14.7	6.1	12.6	5.7	32.3	33.9
car	12.0	11.5	8.4	11.8	5.6	23	7.0	8.8	12	15
machine	7.9	5.5	5.4	7.4	7.4	4.0	4.0	6.0	7.9	11.8
credit	10.1	9.5	3.3	4.9	5.9	7.3	16.9	3.0	5.6	3.0
flags	34.3	19.4	16.6	13.6	9.0	44.0	6.2	13.1	28.0	32.0
hepatitis	10.4	7.0	8.0	5.3	5.3	14.3	3.0	2.0	10.0	8.0
ionosphere	13.6	6.4	6.7	15.8	15.8	7.1	11.9	5.4	18.3	11.6
iris	4.7	3.0	3.3	5.6	5.6	3.6	3.6	3.1	6.3	7
lung-cancer	6.2	3.0	3.2	2.0	2.0	3.7	3.1	2.0	6.3	5.4
pima	20.4	9.7	8.0	12.9	12.9	12.8	53.9	6.2	32.2	25.3
vote	5.2	3.0	3.0	12.8	12.8	18.6	3.2	3.8	11.0	11.2
wave	29.3	9.9	12.1	19.8	19.8	10.2	23.2	9.2	29.0	29.0
wine	6.6	5.0	4.7	8.5	8.5	4.9	8.2	5.9	9.3	6.3
zoo	7.1	6.4	6.0	5.1	5.1	8.4	4.0	5.1	6.8	6.7

TAB. B.2 – *Nombre de règles moyen sur les méthodes implémentées dans SIPINA-W*

généré en revanche, on distingue une série de résultats qui demandent une explication : sur les fichiers "wave", "credit" et surtout "pima", la méthode SIPINA génère un nombre de règle significativement élevé face aux autres méthodes. Ce phénomène s'explique comme suit : durant toute notre expérimentation, afin de mettre les méthodes sur un pied d'égalité, nous n'avons pas mis de contrainte de taille des sommets, donc chaque méthode peut engendrer des feuilles ne contenant qu'un seul individu. Cela ne pose aucun problème pour les arbres, l'expansion s'arrête quand le sommet est pur. Cela est moins vrai pour les graphes, d'autant plus que les paramètres de préférence à la simplicité a été fixé une fois pour toutes sur l'ensemble des fichiers. On constate, dès lors que les classes intègrent un haut niveau de bruit (pour s'en convaincre on remarquera que sur les fichiers pré-cités, CART qui est certainement une stratégie résistant le mieux au bruit construit des petits arbres), SIPINA procède à des regroupements intempestifs qui l'amènent par la suite a des successions de fusions-segmentations. De fait, le nombre de chemins (les règles conjonctives) pour parvenir aux feuilles est élevé. Ce phénomène a également été observé dans la méthode d'induction de graphes fondés sur les MML de [Oliver, 1993].

Bibliographie

[Aczel et Daroczy, 1975] J. Aczel and Z. Daroczy. *On measures of information and their characterizations*. Academic Press, 1975.

[Agrawal *et al.*, 1992] R. Agrawal, S. Ghosh, T. Imielinski, B Iyer, and A. Swami. An interval classifier for database mining applications. In *Proceedings of the 18th International Conference on Very Large Data Bases*, pages 560–573, San Mateo, CA, 1992. Morgan Kaufmann.

[Agrawal et Shafer, 1996] R. Agrawal and J.C. Shafer. Parallel mining of association rules. *IEEE Transaction on Knowledge and Data engineering*, 8(6):962–969, 1996.

[Agresti, 1990] A. Agresti. *Categorical data analysis*. John Wiley, New York, 1990.

[Aivazian *et al.*, 1986] S. Aivazian, I. Enukov, and L. Mechalkine. *Elements de modelisation et traitement primaire des donnees*. MIR, Moscou, 1986.

[Almuallim et Dietterich, 1991] H. Almuallim and T. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 547–552, 1991.

[Almuallim et Dietterich, 1992] H. Almuallim and T. Dietterich. Efficient algorithms for identifying relevant features. In *Proceedings of the 9th Canadian Conference on Artificial Intelligence*, pages 38–45, 1992.

[Andrews *et al.*, 1995] R. Andrews, J. Dietterich, and A.B. Tickle. A survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8:373–389, 1995.

[Asseraf, 1996] M. Asseraf. Extension de la distance de kolmogorov-smirnov et la strategie de la coupure optimale. In *Actes Des 4emes Rencontres de La Societe Francophone de Classification*, 1996.

[Auer *et al.*, 1995] P. Auer, R.C. Holte, and W. Maas. Theory and application of agnostic pac-learning with small decision trees. In *Proceedings of the 12th International Conference on Machine Learning*, pages 21–29. Morgan Kaufmann, 1995.

[Auray *et al.*, 1993] J.P. Auray, G. Duru, M. Lamure, A. Pelc, and A. Zighed. *Guidelines for evaluation in health care economics.* GYD Institute - Lyon, 1993.

[Bahl *et al.*, 1989] L.R. Bahl, P.F. Brown, P.V. De Souza, and R.L. Mercer. A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:1001–1008, 1989.

[Baker et Jain, 1976] F.A. Baker and A.K. Jain. On feature ordering in practice and some finite sample effects. In *Proceedings of the 3rd International Joint Conference on Pattern Recognition*, pages 45–49, San Diego, CA, 1976.

[Baxter et Dowe, 1994] R.A. Baxter and D.L. Dowe. Model selection in linear regression using the mml criterion. In *Proceedings of the 4th IEEE Data Compression Conference*, page 498, 1994.

[Baxter et Oliver, 1994] R.A. Baxter and J.J. Oliver. Mdl and mml : Similarities and differences. Technical Report TR 207, Department of Computer Science - Monash University, 1994.

[Ben-Bassat, 1987] M. Ben-Bassat. Use of distance measures, information measures and error bounds on feature evaluation. In P.R. Krishnaiah and L.N. Kanal, editors, *Classification, Pattern Recognition and Reduction of Dimensionality, Volume 2 of HandBook of Statistics*, pages 773–791. North-Holland Publishing Company, 1987.

[Berckman, 1995] N.C. Berckman. Value grouping for binary decision trees. Technical report, Computer Science Department - University of Massachusetts, April 1995.

[Blumer *et al.*, 1987] A. Blumer, A. Ehrenfeucht, D. Haussler, and M.K. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.

[Bouroche et Tenenhaus, 1970] J.P. Bouroche and M. Tenenhaus. Quelques méthodes de segmentation. *RAIRO, 42*, pages 29–42, 1970.

[Boyen et Wehenkel, 1996] X. Boyen and L. Wehenkel. Automatic induction of continuous decision trees. In *Proceedings of IPMU'96*, 1996.

[Brayton *et al.*, 1992] R.K Brayton, G.D Hatchel, C.T. Mc Mullen, and A.L. Sangiovanni-Vincentelli. *Logic Minimisation Algorithms for VLSI Synthesis.* Kluver Academic, 1992.

[Breiman *et al.*, 1984] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* California : Wadsworth International, 1984.

[Breiman et Shang, 1996] L. Breiman and N. Shang. Born again trees. Technical report, Department of Statistics, University of California at Berkeley, 1996.

[Breiman, 1994] L. Breiman. Bagging predictors. Technical Report 421, Department of Statistics, University of California, 1994.

[Breiman, 1996a] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[Breiman, 1996b] L. Breiman. Bias, variance and arcing classifiers. Technical Report 460, Department of Statistics, University of California at Berkeley, 1996.

[Brodley et Friedl, 1996] C.E. Brodley and M.A. Friedl. Identifying and eliminating mislabeled training instances. In *Proceedings of the 13th National Conference on Artificial Intelligence*, 1996.

[Brunet, 1992] T. Brunet. Le probleme de la resistance aux valeurs inconnues dans l'induction: une foret de branches, 1992.

[Bryant, 1986] R.E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, 1986.

[Bucy et Diesposti, 1993] R.S. Bucy and R.S. Diesposti. Decision tree design by simulated annealing. *Mathematical Modeling and Numerical Analysis*, 27(5):515–534, 1993.

[Bulle, 1990] T. Bulle. *Comparaison de populations - Tests non parametriques et analyse de variance*. Masson, 1990.

[Buntine et Caruana, 1991] W. Buntine and R. Caruana. Introduction to ind and recursive partitionning. Technical Report FIA-91-28, RIACS and NASA Ames Research Center, Moffett Field, CA, 1991.

[Buntine et Niblett, 1992] W. Buntine and T. Niblett. A further comparison of splitting rules for decision tree induction. *Machine Learning*, 8:75–85, 1992.

[Buntine, 1990] W.L. Buntine. Myths and legends in learning classification rules. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 736–742, 1990.

[Buntine, 1991] W. Buntine. *A theory of learning classification rules*. PhD thesis, University of Technology, Sydney, 1991.

[Buntine, 1992] W. Buntine. Learning classification trees. *Statistics and Computing*, 2:63–73, 1992.

[Caperaa et Cutsem, 1988] P. Caperaa and B. Van Cutsem. *Methodes et modeles en statistique non parametrique - Expose fondamental*. Presses de l'universite, Dunod, 1988.

[Carter et Catlett, 1987] C. Carter and J. Catlett. Assessing credit card applications using machine learning. *IEEE Expert, Fall issue*, pages 71–79, 1987.

[Casey et Nagy, 1984] R.G. Casey and G. Nagy. Decision tree design using a probabilistic model. *IEEE Transactions on Information Theory*, IT-30(1):93–99, 1984.

[Catlett, 1991a] J. Catlett. *Megainduction : machine learning on very large databases.* PhD thesis, University of Sydney, 1991.

[Catlett, 1991b] J. Catlett. On changing continuous attributes into discrete ordered discrete attributes. In Y. Kodratoff, editor, *Proceedings of the European Working Session on Learning*, pages 164–178. Springer-Verlag, 1991.

[Celeux et Mkhadri, 1994] G. Celeux and A. Mkhadri. Methodes derivees du modele multinomial. In G. Celeux and J.P. Nakache, editors, *Analyse Discriminante Sur Variables Qualitatives*, chapter 2. Polytechnica, 1994.

[Celeux et Robert, 1993] G. Celeux and C. Robert. Une histoire de discretisation (avec commentaires). *La Revue du Modulad*, pages 7–43, 1993.

[Cestnik *et al.*, 1987a] B. Cestnik, I. Kononenko, and I. Bratko. Assistant-86 : A knowledge elicitation tool for sophisticated users. In I. Bratko and N. Lavrac, editors, *Progress in Machine Learning*. Sigma Press, Bled, Yugoslavia, 1987.

[Cestnik *et al.*, 1987b] B. Cestnik, I. Kononenko, and I. Bratko. ASSISTANT 86: a knowledge-elicitation tool for sophisticated users. In I. Bratko and N. Lavrač, editors, *Progress in Machine Learning*, pages 31–45, Wilmslow, 1987. Sigma Press.

[Cestnik, 1990] B. Cestnik. Estimating probabilities : A crucial task in machine learning. In *ECAI-90*, 1990.

[Chaitin, 1966] G.J. Chaitin. On the length of programs for computing finite sequences. *Journal of the Association for Computing Machinery*, 13:547–549, 1966.

[Chan *et al.*, 1991] C.C. Chan, C. Batur, and A. Srinivasan. Determination of quantization intervals in rule based model for dynamic systems. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, pages 1719–1723, 1991.

[Chandon et Pinson, 1981] J-L. Chandon and S. Pinson. *Analyse typologique - Theories et applications.* Masson, 1981.

[Cheng *et al.*, 1988] J. Cheng, U.M. Fayyad, K.B. Irani, and Z. Qian. Improved decision trees : a generalized version of id3. In *Proceedings of the 5th International Conference on Machine Learning*, pages 100–108, San Mateo, CA, 1988. Morgan Kaufmann.

[Cherkauer et Shavlik, 1996] K.J. Cherkauer and J.W. Shavlik. Growing simpler decision trees to facilitate knowledge discovery. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[Chmielevski et Grzymala-Busse, 1994] M.R. Chmielevski and J.W. Grzymala-Busse. Global discretization of continuous attributes as preprocessing for machine learning. In *Third International Workshop on Rough Sets and Soft Computing*, pages 294–301, 1994.

[Chou, 1988] P.A. Chou. *Applications of Information Theory to Pattern Recognition and the Design of Decision Trees and Trellises*. PhD thesis, Dept. of Electrical Engeneering, Stanford University, 1988.

[Chou, 1991] P.A. Chou. Optimal partitioning for classification and regression trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4), 1991.

[Ciampi *et al.*, 1995] A. Ciampi, E. Diday, J. Lebbe, E. Perinel, and R. Vignes. Tree growing with probabilistically imprecise data. In *Actes de La Conference Internationale Sur L'Analyse Ordinale et Symbolique Des Donnees*, Paris, 1995.

[Clark *et al.*, 1994] P. Clark, B. Cestnik, C. Sammut, and J. Stender. Applications of machine learning : Notes from the panel members. In Y. Kodratoff, editor, *Proceedings of the 5th European Conference on Machine Learning*, pages 457–562. Springer Verlag, 1994.

[Clark et Boswell, 1991] P. Clark and R. Boswell. Rule induction with cn2 : Some recent improvements. In *ML-Proceedings of the 5th European Conference*, pages 151–163, 1991.

[Clark et Niblett, 1989] P. Clark and T. Niblett. The CN2 induction algorithm. 3:261–283, 1989.

[Clark et Pregibon, 1992] L.A. Clark and D. Pregibon. Tree-based models. In J.M. Chambers and T.J. Hastie, editors, *Statistical Models*, pages 377–420. Wadsworth and Brooks, California, 1992.

[Clark, 1990] P. Clark. Machine learning : Techniques and recent developments. In A.R. Mirzai, editor, *Artificial Intelligence : Concepts and Applications in Engineering*, pages 65–93. Chapman and Hall, 1990.

[Cohen et Jensen, 1996] P.R. Cohen and D. Jensen. Overfitting explained. Technical report, Department of Computer Science - University of Massachussetts, 1996.

[Cohen, 1996] W.W. Cohen. Learning trees and rules with set-valued features. Technical report, ATT Laboratories, 1996.

[Corlett, 1983] R.A. Corlett. Explaining induced decision trees. *Expert Systems*, pages 136–142, 1983.

[Cover et Joy, 1991] T.M. Cover and A.T. Joy. *Elements of Information Theory.* John Wiley and Sons, Inc., New York, 1991.

[Cramer, 1946] H. Cramer. *Mathematical methods of statistics.* Princeton University Press, Princeton NJ, 1946.

[Craven et Schavlik, 1996] M. Craven and W. Schavlik. Extracting tree-structured representations of trained networks. *Advances in Neural Information Processing Systems*, 8:24–30, 1996.

[Craven et Shavlik, 1994a] M.W. Craven and J.W. Shavlik. Using sample and queries to extract rules from trained neural networks. In W.W. Cohen and H. Hirsh, editors, *Proceedings of the 11th International Conference on Machine Learning.* Morgan Kaufmann, 1994.

[Craven et Shavlik, 1994b] M.W. Craven and J.W. Shavlik. Using sampling and queries to extract rules from trained neural networks. In *Proceedings of the 11th International Conference on Machine Learning*, pages 37–45, New Brunswick, NJ, 1994. Morgan Kaufmann.

[Cremillieux et Robert, 1996] B. Cremillieux and C. Robert. A pruning method for decision trees in uncertain domains : Applications in medecine. In *Proceedings of the Workshop Intelligent Data Analysis in Medecine and Pharmacology, ECAI'96*, 1996.

[Cremillieux, 1991] B. Cremillieux. *Induction automatique : aspects theoriques, le systeme ARBRE, applications en mdecine.* PhD thesis, Universite Joseph Fourier, Grenoble, 1991.

[D'Alche-Buc *et al.*, 1994] F. D'Alche-Buc, D. Zwierski, and J-P. Nadal. Trio-learning : a new strategy for building hybrid neural trees. *International Journal of Neural Systems*, 5(4):259–274, 1994.

[Danel, 1997] V. Danel. *Des bases de cas vers l'aide a la connaissance - A propos des comas toxiques.* PhD thesis, Universite Joseph Fourier - Grenoble I, 1997.

[Daroczy, 1970] Z. Daroczy. Generalized information function. *Information and control*, 16:36–51, 1970.

[de Merckt et Quinlan, 1996] T. Van de Merckt and J.R. Quinlan. Two-threshold splits of continuous attributes in decision trees. Technical report, The Basser Dept. of Computer Science - University of Sydney, Australia, 1996.

[de Merckt, 1993] T. Van de Merckt. Decision trees in numerical attribute spaces. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1016–1021, 1993.

[Diday *et al.*, 1982] E. Diday, J. Lemaire, J. Pouget, and F. Testu. *Elements d'analyse de donnees.* Dunod, 1982.

[Diday, 1995] E. Diday. Probabilistic objects for a symbolic data analysis. *DIMACS*, 19, 1995.

[Dietterich et Kong, 1995a] T.G. Dietterich and E. Kong. Error-correcting output coding correct bias and variance. In *Proceedings of the 12th International Conference on Machine Learning*, pages 313–321, 1995.

[Dietterich et Kong, 1995b] T.G. Dietterich and E.B. Kong. Machine learning bias, statistical bias and statistical variance of decision trees algorithms. In *Proceedings of the Twelth International Conference on Machine Learning*, 1995.

[Dietterich et Shavlik, 1990] T.G. Dietterich and J.W. Shavlik, editors. *Readings in Machine Learning.* Morgan Kaufmann Publishers, Inc, 1990.

[Dietterich, 1986] T.G. Dietterich. Learning at the knowledge level. *Machine Learning*, 1(3):287–316, 1986.

[Dietterich, 1990] T. G. Dietterich. Machine learning. *Annual Review of Computer Science*, 4:255–306, 1990.

[Dietterich, 1996] T. Dietterich. Statistical tests for comparing supervised learning algorithms. Technical report, Oregon State University, 1996.

[Dietterich, 1997] T.G. Dietterich. Machine learning research : four current directions. Technical report, Department of Computer Science - Oregon State University, 1997.

[DiPalma *et al.*, 1997] S. DiPalma, J. DaRugna, and D.A. Zighed. Apprentissage supervise polythetique : une evaluation. In *Actes Des Cinquiemes Rencontres de La Societe Francophone de Classification*, pages 191–194, 1997.

[Dougherty *et al.*, 1995] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous attributes. In Morgan Kaufmann, editor, *Machine Learning: Proceedings of the Twelth International Conference (ICML-95)*, pages 194–202, 1995.

[Dowe *et al.*, 1992] D.L. Dowe, J. Oliver, L. Allison, C.S. Wallace, and T.I. Dix. A decision graph explanation of protein secondary structure prediction. Technical Report 92/163, Department of Computer Science - Monash University, 1992.

[Dowe et Krusel, 1993] D.L. Dowe and N. Krusel. A decision tree model of bushfire activity. In *Proceedings of the 6th Australian Joint Conference on Artificial Intelligence*, 1993.

[Dravnieks, 1976] A. Dravnieks. Atlas of odor character profiles. *ASTM Data Series*, (DS 61), 1976.

[Efron et Tibshirani, 1993] B. Efron and R. Tibshirani. *An introduction to the bootstrap.* Chapman and Hall, 1993.

[Efron et Tibshirani, 1995] B. Efron and R. Tibshirani. Cross-validation and the bootstrap : Estimating the error rate of a prediction rule. Technical Report 176, Department of Statistics, University of Toronto, 1995.

[Efron, 1983] B. Efron. Estimating the error rate of a prediction rule : improvement on cross-validation. *Journal of American Statistical Association*, 78:316–331, 1983.

[Efron, 1986] B. Efron. How biased is the apparent error rate of a prediction rule? *Journal of American Statistical Association*, 85:79–89, 1986.

[Elomaa, 1994] T. Elomaa. In defense of c4.5 : Notes on learning one-level decision trees. In *Proceedings of the 11th International Conference on Machine Learning*, pages 62–69. Morgan Kaufmann, 1994.

[Esposito *et al.*, 1995] F. Esposito, D. Malerba, and G. Semeraro. Simplifying decision trees by pruning and grafting: New results. *Lecture Notes in Computer Science*, 912:287–??, 1995.

[Everitt, 1977] B.S. Everitt. *The analysis of contingency tables.* Chapman and Hall, London, 1977.

[Fayyad *et al.*, 1996a] U. Fayyad, D. Haussler, and P. Stolorz. Kdd for science data analysis : issues and examples. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[Fayyad *et al.*, 1996b] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Knowledge discovey and data mining : Towards an unifying framework. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[Fayyad et Irani, 1990] U.M. Fayyad and K.B. Irani. What should be minimized in a decision tree? In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 749–754, Boston, MA, 1990. Morgan Kaufmann.

[Fayyad et Irani, 1991] U.M. Fayyad and K.B. Irani. A machine learning algorithm for automated knowledge acquisition : Improvements and extensions. Technical Report CS-634, General Motors Research Labs, 1991.

[Fayyad et Irani, 1992] U.M. Fayyad and K. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8:87–102, 1992.

[Fayyad et Irani, 1993] U.M. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of The 13th Int. Joint Conference on Artificial Intelligence*, pages 1022–1027. Morgan Kaufmann, 1993.

[Fischer, 1958] W.D. Fischer. On grouping for maximum of homogeneity. *Jour. Ann. Statis. Assoc.*, pages 789–798, 1958.

[Fisher et Schlimmer, 1988] D.H. Fisher and J.C. Schlimmer. Concept simplification and prediction accuracy. In *Proceedings of the 5th International Conference in Machine Learning*, pages 22–28. Morgan Kaufmann, 1988.

[Fisher, 1936] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[Fisher, 1992] D. Fisher. Pessimistic and optimistic induction. Technical Report CS-92-12, Department of Computer Science - Vanderbilt University, 1992.

[Flann et Dietterich, 1986] N. Flann and T. Dietterich. Selecting appropriate representation for learning from examples. In *Proceedings of AAAI-86*, pages 460–466. Morgan Kaufmann, 1986.

[Fortune *et al.*, 1996] E. Fortune, P. Makris, and J.P. Asselin de Beauville. Pertinence des indicateurs de bon classement dans le cas d'evenements rares. In *Actes Des Quatriemes Journees de La Societe Francophone de Classification*, 1996.

[Freeman, 1985] T.G. Freeman. Selecting the best model to fit data. *Mathematics and Computers in Simulation*, 27:137–140, 1985.

[Freund et Schapire, 1995] Y. Freund and R.E. Schapire. A decision theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the 2nd European Conference on Computational Learning Theory*, pages 23–37. Springer Verlag, 1995.

[Freund et Schapire, 1996] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

[Friedman, 1977] J.H. Friedman. A recursive partitioning decision rule for nonparametric classifier. *IEEE Transactions on Computers*, C-26:404–408, 1977.

[Friedman, 1996] J. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. Technical report, Department of Statistics and Stanford Linear Accelerator Center, Stanford University, 1996.

[Fulton *et al.*, 1995] T. Fulton, S. Kasif, and S. Salzberg. Efficient algorithms for finding multi-way splits for decision trees. In *Proceedings of the 12th International Conference in Machine Learning*, pages 244–251. Morgan Kaufmann, 1995.

[Furnkranz, 1994] J. Furnkranz. Fossil : A robust relational learner. In *Proceedings of European Conference on Machine Learning*, 1994.

[Ganascia, 1987] J.G. Ganascia. *AGAPE et CHARADE: deux techniques d'apprentissage symbolique appliquees a la construction de bases de connaissances.* PhD thesis, Universite de Paris-Sud, 1987.

[Garey et Graham, 1974] M.R. Garey and R.L. Graham. Performance bounds on the splitting algorithm for binary testing. *Acta Informatica*, 3:347–355, 1974.

[Gavin *et al.*, 1997] G. Gavin, R. Rakotomalala, and D.A. Zighed. Heuristique de ponderation de classifieurs en vue de l'amlioration des performances du bagging. In *Actes Des Cinquiemes Rencontres de La Societe Francophone de Classification*, 1997.

[Gavin, 1997] G. Gavin. Apprentissage supervise : Agregation de predicteurs. Rapport de DEA, DIL (Diplome d'etude approfondie d'Informatique de Lyon), 1997.

[Gelfand *et al.*, 1991] S. B. Gelfand, C. S. Ravishankar, and E. J. Delp. An iterative growing and pruning algorithm for classification tree design. 13:163–174, 1991.

[Georgeff et Wallace, 1984] M.P. Georgeff and C.S. Wallace. A general criterion for inductive inference. In *Proceedings of the 6th European Conference on Artificial Intelligence*, 1984.

[Gillo, 1972] M.W. Gillo. Maid : A honeywell 600 program for an automated survey analysis. *Behavioral Science*, 17:251–252, 1972.

[Gini, 1938] C.W. Gini. Variabilita e mutabilita, contributo allo studio delle distribuzioni e relazioni statische. Technical report, Studi Economico-Giuridici della R. Universita di Caligiari, 1938.

[Giraud, 1993] R. Giraud. *L'Econometrie*. Presses Universitaires de France, 1993.

[Goodman et Kruskal, 1972] L.A. Goodman and W.H. Kruskal. Measures of association for cross classifications iv : simplification of asymptotic variances. *Journal of American Statistical Association*, 67:415–421, 1972.

[Goodman et Kruskall, 1954] L.A. Goodman and W.H. Kruskall. Measures of association for cross classifications. *Journal of American Statistical Association*, 49:732–764, 1954.

[Goodman et Smyth, 1988] R.M. Goodman and P. Smyth. Information-theoretic rule induction. In *Proceedings of European Conference on Artificial Intelligence*, 1988.

[Graham et Hell, Annals of History of Computing] R. Graham and P. Hell. On the history of minimum spanning tree problem. *1985*, 7, Annals of History of Computing.

[Grange *et al.*, 1995] M. Grange, D.A. Zighed, and B. Payri. Knowldege improvement through genetic algorithms. In *Proceedings of the Second Annual Joint Conference on Information Science*, pages 265–266, 1995.

[Gras *et al.*, 1996] R. Gras, F. Guillet, P. Peter, and J. Philippe. Apprentissage automatique et implication : mise en oeuvre sur un espace d'apprentissage en ressources humaines. In *Actes des 4emes Rencontres de la Societe Francophone de Classification*, 1996.

[Gras et Lahrer, 1993] R. Gras and A. Lahrer. L'implication statistique : une nouvelle methode d'analyse de donnees. *Mathematiques Informatique et Sciences Humaines*, 120:5–31, 1993.

[Gras et Ratsimba-Rajohn, 1996] R. Gras and H. Ratsimba-Rajohn. analyse non symetrique des donnees par l'implication statistique. *RAIRO*, 30, 1996.

[Gras, 1979] R. Gras. Contribution a l'etude experimentale et a l'analyse de certaines acquisitions cognitives et de certains objectifs didactiques en mathematiques. Master's thesis, Universite de Rennes 1, 1979.

[Grother, 1992] P.J. Grother. Karhunen loeve feature extraction for neural handwritten character recognition. In *Proceedings of Applications of Artificial Neural Networks III*, volume 1709, pages 155–166, Orlando, April 1992.

[Gueguen, 1994] A. Gueguen. Arbres de discrimination binaire. In G. Celeux and J.P. Nakache, editors, *Analyse Discriminante Sur Variables Qualitatives*, chapter 7. Polytechnica, 1994.

[Guenther, 1966] W.C. Guenther. *Analysis of Variance*. Prentice-Hall, Inc, 1966.

[Hart, 1984] A. Hart. Experience in the use of an inductive system in knowledge engineering. In M. Bramer, editor, *Research and Development in Expert Systems*. Cambridge University Press, 1984.

[Hartmann *et al.*, 1982] C.R.P. Hartmann, P.K. Varshney, K.G. Mehrotra, and C.L. Gerberich. Application of information theory to the construction of efficient decision trees. *IEEE Transactions on Information theory*, IT-28(4):565–577, 1982.

[Heath *et al.*, 1993a] D. Heath, S. Kasif, and S. Salzberg. k-dt : A multi-tree learning method. In *Proceedings of the 2nd International Workshop on Multistrategy Learning*, pages 138–149, 1993.

[Heath *et al.*, 1993b] D. Heath, S. Kasif, and S. Salzberg. Learning oblique decision trees. In *Proceedings of International Joint Conference on Artificial Intelligence - IJCAI'93*, pages 1002–1007, 1993.

[Henrichon et Fu, 1969] E.G. Henrichon and K.S. Fu. A nonparametric partitionning procedure for pattern classification. *IEEE Transactions on Computers*, C-18(7):614–624, 1969.

[Herr *et al.*, 1997] A. Herr, N.I. Klomp, and J.S. Atkinson. Identification of bat echolocation calls using a decision tree classification system. *Complexity International*, 4, 1997. http://www.cs.edu.au/ci/vol4/herr/batcall.html.

[Ho et Nguyen, 1996] T. Ho and T. Nguyen. Evaluation of attribute selection measures in decision tree induction. Technical report, Japan Advanced Institute of Science and Technology, 1996.

[Holsheimer *et al.*, 1996] M. Holsheimer, M. Kersten, and A. Siebes. Data surveyor: searching the nuggets in parallel. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*. AAAI Press, 1996.

[Holsheimer et Siebes, 1994] M. Holsheimer and A. Siebes. Data mining : The search for knowledge in databases. Technical Report CS-R9406, Department of Algorithmics and Architecture - Centrum voor Wiskunde en Informatica, 1994.

[Holte, 1993] R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.

[Hughes, 1968] G.F. Hughes. On the mean accuracy of statistical pattern recognition. *IEEE Trans. Info. Theory*, IT-14(1):55–63, 1968.

[Hunt *et al.*, 1966] E.B. Hunt, J. Marin, and P.J. Stone. *Experiments in Induction*. Academic Press, New York, 1966.

[Hyafil et Rivest, 1976] L. Hyafil and R.L. Rivest. Constructing optimal binary decision tree is np-complete. *Information Processing Letters*, 5:15–17, 1976.

[Iba et Langley, 1992] W.F. Iba and P. Langley. Induction of one-level decision trees. In *Proceedings of the 9th International Conference on Machine Learning*, pages 233–240. Morgan Kaufmann, 1992.

[Imam, 1994] I.F. Imam. An empirical study on the incompetence of attribute selection criteria. Technical report, Machine Learning and Inference Laboratory - George Mason University, 1994.

[Imam, 1995] I.F. Imam. An empirical study on the incompetence of attribute selection criteria. Technical report, Machine Learning and Inference Laboratory - Greorge Mason University, 1995.

[Jacquard, 1992] A. Jacquard. *Les probabilits*. Presses Universitaires de France, 1992.

[Jain *et al.*, 1987] A.K. Jain, R.C. Dubes, and C. Chen. Bootstrap techniques for error estimation. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-9(5):628–633, 1987.

[Janikow, 1993] C.Z. Janikow. Fuzzy processing in decision trees. In *Proceedings of the International Symposium on Artificial Intelligence*, pages 360–367, 1993.

[Jaynes, 1994] E.T. Jaynes. *Probability theory: the logic of science*. 1994. edition incomplete de Juin 94.

[John, 1994] G.H. John. Cross-validated c4.5 : Using error estimation for automatic parameter selection. Technical Report STAN-CS-TN-94-12, Computer Science Department, Stanford University, 1994.

[John, 1995] G.H. John. Robust decision trees : Removing outliers from databases. In *Proceedings of the 1st International Conference on Knowledge Discovery Data Mining*, pages 174–179, 1995.

[Kalkanis, 1993] G. Kalkanis. The application of confidence interval error analysis to the design of decision tree classifiers. *Pattern Recognition Letters*, 14(5):355–361, 1993.

[Kass, 1980] G.V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2):119–127, 1980.

[Kerber, 1992] R. Kerber. Discretization of numeric attributes. In MIT Press, editor, *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 123–128, 1992.

[Kervahut et Potvin, 1996] T. Kervahut and J-Y. Potvin. An interactive-graphic environment for automatic generation of decision trees. *Decision Support Systems*, 18:117–134, 1996.

[Kira et Rendell, 1992] K. Kira and L. Rendell. A practical approach to feature selection. In *Proc. Int. Conf. In Machine Learning*, pages 249–256, 1992.

[Knoll *et al.*, 1994] U. Knoll, G. Nakhaeizadeh, and B. Tausend. Cost-sensitive pruning of decision trees. *Lecture Notes in Computer Science*, 784:383–??, 1994.

[Kodratoff et Manago, 1987] Y. Kodratoff and M. Manago. Generalization and noise. *International Journal of Man-Machine Studies*, 27:181–204, 1987.

[Kodratoff, 1997] Y. Kodratoff. L'extraction de connaissances a partir de donnees : un nouveau sujet pour la recherche scientifique. *Revue electronique R.E.A.D*, 1997.

[Kohavi *et al.*, 1994] R. Kohavi, G. John, R. Long, D. Manley, and K. Pfleger. Mlc++: A machine learning library in c++. *Tools with Artificial Intelligence*, pages 740–743, 1994.

[Kohavi et John, 1997] R. Kohavi and G. John. Wrappers for feature subset selection. *Journal of Artificial Intelligence, Special issue on Relevance*, 1997.

[Kohavi et Kunz, 1997] R. Kohavi and C. Kunz. Option decision trees with majority votes. In *Proceedings of the International Conference on Machine Learning - ICML'97*, 1997.

[Kohavi et Li, 1995] R. Kohavi and C.H. Li. Oblivious decision trees, graphs and top-down pruning. In *Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI'95*, 1995.

[Kohavi et Sahami, 1996] R. Kohavi and M. Sahami. Error-based and entropy-based discretization of continuous features. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, Menlo-Park, 1996.

[Kohavi et Wolpert, 1996] R. Kohavi and D.H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996.

[Kohavi, 1994] R. Kohavi. Bottom-up induction of oblivious read-once decision graphs. In *Proceedings of the European Conference on Machine Learning*, 1994.

[Kohavi, 1995a] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI'95*, 1995.

[Kohavi, 1995b] R. Kohavi. *Wrappers for performance enhancement and oblivious decision graphs*. PhD thesis, Department of Computer Science - Stanford University, 1995.

[Kolmogorov, 1965] A.N. Kolmogorov. Three approaches to the quantitative defintion of information. *Problems of Information Transmission*, 1:4–7, 1965.

[Kononenko, 1992] I. Kononenko. Combining decisions of multiple rules. In B. Du Boulay and V. Sgurev, editors, *Artificial Intelligence V: Methodlogy, Systems, Applications*, pages 87–96. Elsevier Science, Amsterdam, 1992.

[Kononenko, 1993] I. Kononenko. Inductive and bayesian learning in medical diagnosis. *Applied Artificial Intelligence*, 7:317–337, 1993.

[Kononenko, 1994] I. Kononenko. Estimating attributes : Analysis and extension of relief. In *Proc. European Conf. On Machine Learning - ECML'94*, pages 171–182, 1994.

[Kononenko, 1995] I. Kononenko. On biases in estimating multi-valued attributes. In *Proc. Int. Joint Conf. On Artificial Intelligence IJCAI'95*, pages 1034–1040, 1995.

[Koza, 1991] J.R. Koza. Concept formation and decision tree induction using the genetic programming paradigm. In H.P. Schwefel and R. Manner, editors, *Parallel Problem Solving from Nature - Proceedings of the 1st Workshop, PPSN 1, Volume 496 of Lecture Notes in Computer Science*, pages 124–128, 1991.

[Krall *et al.*, 1996] S. Krall, M.T. Zubrow, and M. Mitchell. Success in using non-invasive mechanical ventilation is predicted by patient pathophysiology : A retrospective review of 110 patients. *Chest*, 4:110–148, 1996.

[Kvalseth, 1987] T.O. Kvalseth. Entropy and correlation : Some comments. *IEEE Transaction on Systems, Man and Cybernetics*, SMC-17(3):517–519, 1987.

[Kwok et Carter, 1990] S.W. Kwok and C. Carter. Multiple decision trees. In R.D. Schachter, T.S. Levitt, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 327–335. Elsevier Science Publishers, 1990.

[Lagacherie et Holmes, 1996] P. Lagacherie and S. Holmes. Addressing geographical data errors in a classification tree for soil units prediction. *International Journal of Geographical Information System*, 1996.

[Langley et Sage, 1994] P. Langley and S. Sage. Oblivious decision trees and abstract cases. In *Working Notes of the AAAI-94, Workshop on Case-Based Reasoning*, 1994.

[Lechevallier, 1990] Y. Lechevallier. Recherche d'une partition optimale sous contrainte d'ordre totale. Technical Report 1247, INRIA, Juin 1990.

[Lerman *et al.*, 1981] I.C. Lerman, R. Gras, and H. Rostam. Elaboration et evaluation d'un indice d'implication pour donnees binaires. *Mathematiques et Sciences Humaines*, 74:5–35, 1981.

[Lerman et Costa, 1996] I.C. Lerman and F.P. Da Costa. Coefficients d'association et variables a tres grand nombre de categories dans les arbres de decision : application a l'identification de la structure secondaire d'une proteine. Technical Report 2803, INRIA, Fevrier 1996.

[Lerman et Peter, 1985] I.C. Lerman and P. Peter. Elaboration et logiciel d'un indice de similarite entre objets d'un type quelconque. application au problme du consensus en classification. Technical Report 262, Irisa - Rennes, 1985.

[Lerman, 1992a] I.C. Lerman. Conception et analyse de la forme limite d'une famille de coefficients statistiques d'association entre variables relationnelles i. *Rev. Math. Info. Sci. Hum.*, 118:35–52, 1992.

[Lerman, 1992b] I.C. Lerman. Conception et nanlyse de la forme limite d'une famille de coefficients statistiques d'association entre variables relationnelles ii. *Rev. Math. Info. Sci. Hum.*, 119:75–100, 1992.

[Light et Margolin, 1971] R.J. Light and B.H. Margolin. An analysis of variance for categorical data. *Applied Statistics*, 66(335), 1971.

[Liu et Setiono, 1995] H. Liu and R. Setiono. Chi2 : Feature selection and discretization of numeric attributes. In *Proceedings of the 7th IEEE International Conference on Tools with Artificial Intelligence*, 1995.

[Liu et Setiono, 1996] H. Liu and R. Setiono. Dimensionality reduction via discretization. Technical report, Department of Information Systems and Computer Science - National University of Singapore, 1996.

[Lubinsky, 1993] D. Lubinsky. Algorithm speedups in growing classification trees by using an additive split criterion. In *Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics*, pages 435–444, 1993.

[Lubinsky, 1994] D. Lubinsky. *Bivariate splits and consistent split criteria in dichotomous classification trees.* PhD thesis, Department of Computer Science, Rutgers University, New Brunswick, 1994.

[Maas, 1994] W. Maas. Efficient agnostic pac-learning with simple hypothesis. In *Proceedings of the 7th Annual ACM Conference on Computationnal Learning*, pages 67–75, 1994.

[Magerman, 1994] D.M. Magerman. *Natural language sparsing as statistical pattern recognition.* PhD thesis, Stanford University, 1994.

[Mahoney et Mooney, 1991] J.J. Mahoney and R.J. Mooney. Initializing id5r with a domain theory : some negative results. Technical Report 91-154, Computer Science Department - University of Texas at Austin, 1991.

[Mantaras, 1991] R.L. De Mantaras. A distance-based attributes selection measures for decision tree induction. *Machine Learning*, 6:81–92, 1991.

[Marcotorchino, 1984] F. Marcotorchino. Utilisation des comparaisons par paires en statistique des contingences. Technical Report F 071, Centre Scientifique IBM-France, Octobre 1984.

[Margineantu et Dietterich, 1997] D.D. Margineantu and T.G. Dietterich. Pruning adaptive boosting. In *Proceedings of the International Conference on Machine Learning - ICML'97*, 1997.

[Marsala, 1994] C. Marsala. Arbres de decision et sous-ensembles flous. Technical report, Laforia - Universite Paris VI, 1994.

[Marsala, 1996] C. Marsala. Fuzzy partitioning using mathematical morphology in learning scheme. In *Proceedings of the FUZZ'IEEE Conference*, New Orleans, 1996.

[Matheus et Rendell, 1989] C.J. Matheus and L.A. Rendell. Constructive induction on decision trees. In *Proceedings Od IJCAI'89*, pages 645–650, 1989.

[Matheus, 1990] C.J. Matheus. Adding domain knowledge to sbl through feature construction. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 803–808, 1990.

[McKenzie et Low, 1992] D.P. McKenzie and L.H. Low. The construction of computerized classification systems using machine learning. *Computers in Human Behaviour*, 8(2/3):155–167, 1992.

[Mehta *et al.*, 1995] Manish Mehta, Jorma Rissanen, and Rakesh Agrawal. MDL-based decision tree pruning. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, pages 216–221, 1995.

[Meisel et Michalopoulos, 1973] W.S. Meisel and D.A. Michalopoulos. A partitionning algorithm with application in pttern classification and the optimization of decision trees. *IEEE Transactions on Computers*, C-22(1):93–103, 1973.

[Michalski *et al.*, 1986] R.S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In *Proceddings of AAAI-86*, pages 1041–1045, Philadelphia, PA, 1986.

[Michalski et Imam, 1994] R.S. Michalski and I. Imam. Learning problem-oriented dcision structures from decsion rules : the aqdt-2 system. *Lecture notes in Artificial Intelligence : Methodology for Intelligent systems*, (869):416–426, 1994.

[Michalski et Larson, 1983] R.S. Michalski and J. Larson. Selection of most representative training examples and incremental generation of vl1 hypothesis. Technical Report ISG 83-5, Department of computer science, University of Illinois at Urbana Champaign, Urbana, 1983.

[Michalski, 1969] R.S. Michalski. On the quasi-minimal solution of the general covering problem. In *Proceedings of the 5th International Symposium on Information Processing*, pages 125–128, Bled Yugoslavia, 1969.

[Michalski, 1983] R.S. Michalski. Theory and methodology of inductive learning. In R.S. Michalski, J.G. Carbonnel, and T.M. Mitchell, editors, *Machine Learning : An Artificial Intelligence Approach*, volume 1, pages 83–134. Morgan Kaufmann, Los Altos, 1983.

[Michie, 1979] D. Michie. *Expert Systems in the Micro Electronic age*. Edimburgh University Press, 1979.

[Michie, 1987] D. Michie. Current developments in expert systems. *Applications of Expert Systems*, 1:137–156, 1987.

[Mingers, 1987] J. Mingers. Expert systems - rule induction with statistical data. *Journal of the Operational Research Society*, 38:39–47, 1987.

[Mingers, 1989a] J. Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 4:227–243, 1989.

[Mingers, 1989b] J. Mingers. An empirical comparison of selection measures for decision tree induction. *Machine Learning*, 3:319–342, 1989.

[Miyakawa, 1989] M. Miyakawa. Criteria for selecting a variable in the construction of efficient decision trees. *IEEE Transactions on Computers*, 38(1):130–141, 1989.

[Mood, 1940] A.M. Mood. The distribution theory of runs. *Ann. of Math. Stat.*, 11:367–392, 1940.

[Morgan et Sonquist, 1963] J. N. Morgan and J. A. Sonquist. Problems in the analysis of survey data, and a proposal. 58:415–434, 1963.

[Mowforth, 1986] P. Mowforth. Some applications with inductive expert systems. Technical Report TIOP 86-002, Turing Institute, Glasgow, UK, 1986.

[Munteanu, 1996] P. Munteanu. *Extraction de connaissances dans les bases de donnees paroles : Apport de l'apprentissage symbolique*. PhD thesis, Institut de la Communication Parlee - Universite de Grenoble, 1996.

[Murphy et Aha, 1995] P.M. Murphy and D.W. Aha. Uci repository of machine learning databases, 1995. Available at http:/www.ics.uci.edu/ mlearn/MLRepository.html.

[Murphy et McCraw, 1991] O.J. Murphy and R.L. McCraw. Designing storage efficient decision trees. *IEEE Trans. on Comp.*, 40(3):315–319, 1991.

[Murphy et Pazzani, 1991] P. Murphy and M. Pazzani. Id2-of-3 : Constructive induction of m-of-n concepts for discriminators in decision trees. Technical Report 91-37, Department of Information and Computer Science - University of California at Irvine, 1991.

[Murphy et Pazzani, 1994] P.M. Murphy and M.J. Pazzani. Exploring the decision forest : An empirical evaluation of occam's razor in decision tree induction. *Journal of Artificial Intelligence Research*, 1:257–275, 1994.

[Murphy, 1995] P.M. Murphy. The benefit of decision tree size biaises as a function of concept distribution. Technical Report TR95-29, Dept. of Information and Computer Science, University of California, Irvine, 1995.

[Murthy *et al.*, 1994] S.K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–33, 1994.

[Murthy et Salzberg, 1992] S.K. Murthy and S. Salzberg. Clustering astronomical objects using minimum spanning trees. Technical report, Deptartment of Computer Science - John Hopkins University, 1992.

[Murthy et Salzberg, 1995a] S.K. Murthy and S. Salzberg. Decision trees induction : How effective is the greedy heuristic? In *Proceedings of the First International Conference on Knowledge Discovery in Databases*, Montreal, Canada, 1995.

[Murthy et Salzberg, 1995b] S.K. Murthy and S. Salzberg. Lookahead and pathology in decision tree induction. In *Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI'95*, 1995.

[Murthy, 1995] S.K. Murthy. *On Growing Better Decision Trees from Data.* PhD thesis, John Hopkins University, 1995.

[Mutchler, 1993] D. Mutchler. The multi-player version of minimax displays game pathology. *Artificial Intelligence*, 64(2):323–336, 1993.

[Nau, 1983] D.S. Nau. Decision quality as a function of search on game trees. *Journal of the ACM*, 30(4):687–708, 1983.

[Niblett, 1987] T. Niblett. Constructing decision trees in noisy domains. In I. Bratko and N. Lavrac, editors, *Progress in Machine Learning*. Sigma Press, 1987.

[Norton, 1989] S.W. Norton. Generating better decision trees. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 800–805, 1989.

[Nunez, 1988] M. Nunez. Economic induction : a case study. In *Proceedings of the 3rd European Working Session in Learning*, pages 139–145, London, 1988. Pitman.

[Oates et Jensen, 1997] T. Oates and D. Jensen. The effects of training set size on decision tree complexity. In *Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics*, 1997.

[O'Brien et Dyck, 1985] P.C. O'Brien and P.J. Dyck. A runs test based on run lengths. *Biometrics*, 41:237–244, 1985.

[Oliveira et Sangiovanni-Vincentelli, 1995] A.L. Oliveira and A. Sangiovanni-Vincentelli. Using the minimum description length principle to infer reduced ordered decision graphs. *Machine Learning*, 12:1–30, 1995.

[Oliveira et Vincentelli, 1993] A.L. De Oliveira and A.S. Vincentelli. Learning complex boolean functions : Algorithms and applications. In *Advances in Neural Information Processing Systems 6*, pages 911–918. Morgan Kaufmann, Denver, 1993.

[Oliveira, 1994] A.L. Oliveira. *Inductive Learning by Selection of Minimal Complexity Representation.* PhD thesis, University of California at Berkeley, 1994.

[Oliver *et al.*, 1992] J.J. Oliver, D.L. Dowe, and C.S. Wallace. Inferring decision graphs using the minimum message length principle. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, pages 361–367, 1992.

[Oliver et Dowe, 1995] J.J. Oliver and D.L. Dowe. Using unsupervised learning to assist supervised learning. Technical Report 95/235, Department of Computer Science - Monash University, September 1995.

[Oliver et Hand, 1995] J. Oliver and D. Hand. On pruning and averaging decision trees. In A. Prieditis and S. Russel, editors, *Proceedings of the Twelth International Conference on Machine Learning*, pages 430–437. Morgan Kaufmann, 1995.

[Oliver et Wallace, 1991] J.J. Oliver and C.S. Wallace. Inferring decision graphs. In *Proceedings of Workshop 8 - Evaluating and Changing Representation in Machine Learning*, 1991.

[Oliver, 1993] J.J. Oliver. Decision graphs - an extension of decision trees. In *Proceedings of the 4th International Workshop on Artificial Intelligence and Statistics*, pages 343–350, 1993.

[Olszak et Ritschard, 1995] M. Olszak and G. Ritschard. The behaviour of nominal and ordinal partial association measures. *The statistician*, 44(2):195–212, 1995.

[Olszak, 1995] M. Olszak. *Modelisation des relations de causalite entre variables qualitatives.* PhD thesis, Departement d'econometrie - Universite de Geneve, 1995.

[Pagallo et Haussler, 1990] G. Pagallo and D. Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, pages 71–99, 1990.

[Pao, 1989] Y.H. Pao. *Adaptive pattern recognition and neural networks.* Addison Wesley, 1989.

[Patte *et al.*, 1982] F. Patte, M. Etcheto, and P. Laffort. Solubility factors for 240 solutes and 207 stationary phases in gas liquid chromatography. *IEEE*, 54(13):2239–2247, 1982.

[Piasta et Lenarcik, 1997] Z. Piasta and A. Lenarcik. Rule induction with probabilistic rough classifiers, 1997. Communication acceptee dans la revue Machine Learning.

[Picard, 1972] C.F. Picard. *Graphes et questionnaires*. Guathier-Villars, 1972.

[Prechelt, 1996] L. Prechelt. A quantitative study of experimental evaluation of neural network learning algorithm. *Neural Networks*, 9, 1996.

[Press *et al.*, 1988] W.H. Press, B.P. Falnnery, S.A. Teutolsky, and W.T. Vetterling. *Numerical recipes in C*. Cambridge University Press, 1988.

[Quinlan et Rivest, 1989] J.R. Quinlan and R.L. Rivest. Inferring decision trees using the minimum description length. *Information and Computation*, 80:227–248, 1989.

[Quinlan, 1979] J. R. Quinlan. Discovering rules by induction from large collections of examples. In D. Michie, editor, *Expert Systems in the Microelectronic Age*, pages 168–201, Edinburgh, 1979. Edinburgh University Press.

[Quinlan, 1986a] J.R. Quinlan. The effect of noise on concept learning. In R.S. Michalski, J.G. Carbonnel, and T.M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, volume 2, chapter 6. Morgan Kaufmann, 1986.

[Quinlan, 1986b] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[Quinlan, 1987a] J. R. Quinlan. Generating production rules from decision trees. In J. McDermott, editor, *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (Milan, 1987)*, San Mateo, CA, 1987.

[Quinlan, 1987b] J.R. Quinlan. Decision trees as a probabilistic classifiers. In *Proceedings of the 4th International Workshop on Machine Learning*, 1987.

[Quinlan, 1987c] J.R. Quinlan. Simplifying decision trees. In B. Gianes and J. Boose, editors, *Knowledge Acquisition for Knowledge-Based Systems*, pages 239–252. Academic Press, London, 1987.

[Quinlan, 1988a] J.R. Quinlan. Decision trees and multi-valued attributes. *Machine Intelligence*, 11:305–318, 1988.

[Quinlan, 1988b] J.R. Quinlan. An empirical comparison of genetic and decision tree classifiers. In *Proceedings of the 5th International Conference on Machine Learning*, pages 135–141, 1988.

[Quinlan, 1989] J.R. Quinlan. Unknow attribute values in induction. In *Proceedings of the 6th International Workshop on Machine Learning*, pages 164–168, 1989.

[Quinlan, 1990] J. R. Quinlan. Decision trees and decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 20:339–346, 1990.

[Quinlan, 1993a] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[Quinlan, 1993b] J.R. Quinlan. Comparing connectionist and symbolic learning methods. In S. Hanson, G. Drastal, and R. Rivest, editors, *Computational Learning Theory and Natural Learning Systems : Constraints and Prospects*. MIT Press, 1993.

[Quinlan, 1994] J.R. Quinlan. The minimum description length principle and categorical theories. In *Proceedings of the 11th International Conference on Machine Learning*, pages 233–241, 1994.

[Quinlan, 1995] J.R. Quinlan. Mdl and categorical theories (continued). In *Proceedings of the Twelth International Conference on Machine Learning*, pages 464–470. Morgan Kaufmann, 1995.

[Quinlan, 1996] J.R. Quinlan. Bagging, boosting and c4.5. In *Proceedings of the 13th American Association National Conference on Artificial Intelligence*, pages 725–730, Menlo Park, CA, 1996. AAAI Press.

[Rabaseda *et al.*, 1995] S. Rabaseda, R. Rakotomalala, and M. Sebban. Discretization of continuous attributes : a survey of methods. In *Proceedings of the 2nd Annual Joint Conference on Information Sciences*, pages 164–166, 1995.

[Rabaseda *et al.*, 1996a] S. Rabaseda, R. Rakotomalala, and D.A. Zighed. Rules extracted automatically by induction. In *Proceedings of the 6th Conference on Information Processing and Management of Uncertainty*, pages 551–556, 1996.

[Rabaseda *et al.*, 1996b] S. Rabaseda, M. Sebban, and R. Rakotomalala. A comparison of some contextual discretization methods. *Information Sciences : Intelligent Systems*, 1996.

[Rabaseda, 1996] S. Rabaseda. *Contributions a l'extraction automatique de connaissances : application a l'analyse clinique de la marche*. PhD thesis, Universite Claude Bernard - Lyon 1, 1996.

[Ragavan *et al.*, 1993a] H. Ragavan, L. Rendell, M. Shaw, and A. Tessmer. Complex concept acquisition through directed search and feature catching. In *Proceedings of the 13th Internationla Joint Conference on Artificial Intelligence*, pages 946–951, 1993.

[Ragavan *et al.*, 1993b] H. Ragavan, L. Rendell, M. Shaw, and A. Tessmer. Learning complex real-world concepts through feature construction. Technical Report UIUC-BI-AI-DSS-93-01, The University of Illinois at Urbana-Champaign, 1993.

[Ragavan et Rendell, 1991] H. Ragavan and L. Rendell. Relieving limitations of empirical algorithms. In *Change of Representation Workshop - Proceedings of the 12th IJCAI*, 1991.

[Rakotomalala *et al.*, 1996] R. Rakotomalala, D.A. Zighed, and S. Rabaseda. Validation of rules issued from induction graphs. In *Proceedings of the 6th Conference on Information Processing and Management of Uncertainty*, pages 1259–1264, 1996.

[Rakotomalala *et al.*, 1997a] R. Rakotomalala, D.A. Zighed, and F. Feschet. Caracterisation des regles de production dans un processus d'induction. *Soumis a la revue electronique READ*, 1997.

[Rakotomalala *et al.*, 1997b] R. Rakotomalala, D.A. Zighed, and F. Feschet. Rule characterization in induction process. In *Indo-French Workshop on Symbolic Data Analysis and its Applications*, pages 190–199, 1997.

[Rakotomalala et Chettouh, 1996] R. Rakotomalala and E. Chettouh. Extraction de regles par validation dans les graphes d'induction. In *Actes des 4emes journees de la Societe Francophone de Classification*, 1996.

[Rakotomalala et Zighed, 1996] R. Rakotomalala and D.A. Zighed. Evaluation et validation des sommets dans les graphes d'induction : une alternative a l'elagage. In *Actes du XX-Veme Colloque International l'Association Rhodanienne pour l'Avancement de l'Econometrie - ARAE'96*, 1996.

[Rakotomalala et Zighed, 1997] R. Rakotomalala and D.A. Zighed. Mesures d'association dans les graphes d'induction : une appoche statistique de l'arbitrage generalite-precision. In *Proceedings of the 7th Conference of International Association for the Development of Interdisciplinary Research*, pages 131–134, 1997.

[Rakotomalala, 1995a] R. Rakotomalala. Generation et simplification des regles dans le logiciel sipina. Seminaire du laboratoire ERIC, Universite Lumiere Lyon 2 - Septembre 95, 1995.

[Rakotomalala, 1995b] R. Rakotomalala. La distribution theorique des sequences. Technical report, Laboratoire ERIC, Universite Lumiere Lyon 2, 1995.

[Ramdani, 1994] M. Ramdani. *Systeme d'induction formelle a base de connaissances imprecises.* PhD thesis, Institut Blaise Pascal, Universite Paris VI, 1994.

[Rauber *et al.*, 1994] T.W. Rauber, D. Coltuc, and A.S. Steiger-Garcao. Multivariate discretization of continuous attributes for machine learning. Technical report, Universidade Nova de Lisboa - Faculdade de Cienca e Tecnologia, 1994.

[Rauber et Steiger-Garcao, 1993] T.W. Rauber and A.S. Steiger-Garcao. Feature selection of categorical attributes based on contingency table analysis. In *Proceedings of the 5th Portuguese Conference on Pattern Recognition*, 1993.

[Rauber et Steiger-Garccedillo, 1993] T.W. Rauber and M. Steiger-Garccedillo. Decision trees for symbolic knowledge based on contingency table analysis. In *Proceedings of SPIE Int. Symposium on Optical Engineering and Photonics in Aerospace and Remote Sensing*, Orlando, USA, 1993.

[Rendell et Seshu, 1990] L. Rendell and R. Seshu. Learning hard concepts through constructive induction : framework and rationale. *Computational Intelligence*, 6(4):247–270, 1990.

[Rendell, 1986] L. Rendell. A general framework for induction and a study of selective induction. *Machine Learning*, 1(2):177–226, 1986.

[Richeldi et Rossotto, 1995] M. Richeldi and M. Rossotto. Class-driven statistical discretization of continuous attributes. In *Proceedings of European Conference on Machine Learning*, pages 335–338, 1995.

[Rissanen, 1978] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[Rivest, 1987] R. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1987.

[Rounds, 1980] E.M. Rounds. A combined nonparametric approach to feature selection and binary decision tree design. *Pattern Recognition*, 12:313–317, 1980.

[Rubiello, 1997] L. Rubiello. *Techniques innovantes en informatique*. Hermes, 1997.

[Russel et Norvig, 1995] S. Russel and P. Norvig. *Artificial Intelligence : A Modern Approach*. Prentic Hall, 1995.

[Safavian et Landgrebe, 1991] S. R Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man and Cybernetics*, 21:660–674, 1991.

[Salzberg, 1997] S.L. Salzberg. On comparing classifiers : A critique of current research and methods. *Data mining and Knowledge discovery*, 1:1–12, 1997.

[Saporta, 1975] G. Saporta. *Liaisons entre plusieurs ensembles de variables et codage de donnees qualitatives*. PhD thesis, 1975.

[Sarkar *et al.*, 1994] U.K. Sarkar, P.P. Chakrabarti, S. Ghose, and S.C. DeSarkar. Improving greedy algorithms by lookahead-search. *Journal of Algorithms*, 16(1):1–23, 1994.

[Schaffer, 1991] C. Schaffer. When does overfitting decrease prediction accuracy in induced decision trees and rule sets? In *Proceedings of the European Working Session on Learning*, pages 192–205, 1991.

[Schaffer, 1992] C. Schaffer. Sparse data and the effect of overfitting avoidance in decision tree induction. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 147–152, 1992.

[Schaffer, 1993a] C. Schaffer. Overfitting avoidance as a bias. *Machine Learning*, 10:153–178, 1993.

[Schaffer, 1993b] C. Schaffer. Selecting a classification method by cross-validation. *Machine Learning*, 13(1):135–143, 1993.

[Schaffer, 1994] C. Schaffer. A conservation law for generalization performance. In *Proceedings of the 11th International Conference on Machine Learning*, pages 259–265, 1994.

[Schapire et Freund, 1996] R. Schapire and Y. Freund. Experiments with a new boosting algorithm. Technical report, ATT Labs, 1996.

[Scheffer et Herbrich, 1997] T. Scheffer and H. Herbrich. Unbiased assessment of learning algorithm. In *Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI'97*, 1997.

[Sebag, 1995] M. Sebag. 2nd order understability of disjunctive version spaces. In *Workshop of the 14th International Joint Conference on Artificial Intelligence*, 1995.

[Sebban, 1996] M. Sebban. *Modeles theoriques en Reconnaissance de Formes et Architecture Hybride pour Machine Perceptive*. PhD thesis, Universite Claude Bernard - Lyon 1, 1996.

[Sethi et Chatterjee, 1977] I.K. Sethi and B. Chatterjee. Efficient decision tree design for discrete variable pattern recognition. *Pattern recognition*, 9:197–206, 1977.

[Shahshahani et Landgrebe, 1994] B.M. Shahshahani and D.A. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5):1087–1095, 1994.

[Shang et Breiman, 1996] N. Shang and L. Breiman. Distribution based trees are more accurate. In *Proceedings of ICONIP*, 1996.

[Shannon et Weaver, 1949] Claude E. Shannon and Warren Weaver. *The mathematical theory of communication.* University of Illinois Press, 1949.

[Shavlik *et al.*, 1991] J.W. Shavlik, R. Mooney, and G.G. Towell. Symbolic and neural learning algorithms : An experimental comparison. *Machine Learning*, 6, 1991.

[Siegel, 1956] S. Siegel. *Nonparametric Statistics.* McGraw-Hill, New York, 1956.

[Simon, 1983] H. Simon. Why should machines learn ? In R.S. Michalski, J.G. Carbonnel, and T.M. Mitchell, editors, *Machine Learning : An Artificial Intelligence Approach.* Morgan Kaufmann, 1983.

[Smyth et Goodman, 1992] P. Smyth and R. M. Goodman. An information theoretic approach to rule induction from databases. *Ieee Trans. On Knowledge And Data Engineering*, 4:301–316, 1992.

[Solomonoff, 1964] R.J. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7:1–22, 1964.

[Sonquist *et al.*, 1971] J.A. Sonquist, E.L. Baker, and J.N. Morgan. *Searchning for structure.* Institute for Social Research, University of Michigan, 1971.

[Sorkin, 1983] R. Sorkin. A quantitative occam's razor. *International Journal of Theoretical Physics*, 22:109–113, 1983.

[Stone, 1974] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, B 36:111–147, 1974.

[Takenaga et Yajima, 1993] Y. Takenaga and S. Yajima. Np-complteness of minimum binary decision diagram identification. Technical Report COMP 92-99, Institute of Electronics, Information and Communication Engineers of Japan, 1993.

[Tan et Schlimmer, 1990] M. Tan and J. Schlimmer. Two case studies in cost-sensitive concept acquisition. In *Proceedings of AAAI-90, the 8th National Conference on Artificial Intelligence*, pages 854–860. MIT Press, 1990.

[Taylor et Silverman, 1993] P.C. Taylor and B.W. Silverman. Block diagrams and splitting criteria for classification trees. *Statistics and Computing*, 3(4):147–161, 1993.

[Terrenoire, 1970] M. Terrenoire. *Un modele mathematique de processus d'interrogation : les pseudoquestionnaires.* PhD thesis, Universite de Grenoble, 1970.

[Theil, 1970] H. Theil. On the estimation of relationships involving qualitative variables. *American Journal of Sociology*, 76:103–154, 1970.

[Tibshirani, 1996] R. Tibshirani. Bias, variance and prediction error for classification rules. Technical report, Department of preventive Medecine and Biostatistics and Department of Statistics, University of Toronto, 1996.

[Tounissoux, 1980] D. Tounissoux. Processus sequentiels adaptatifs de r.d.f pour l'aide au diagnostic. Master's thesis, Universite Claude Bernard - Lyon 1, 1980.

[Towell *et al.*, 1990] G. Towell, J. Shavlik, and M. Noordwier. Refinement of approximate domain theory by knowledge-based neural networks. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 861–866. Morgan-Kaufmann, 1990.

[Tschuprow, 1921] A.A. Tschuprow. On the mathematical expectation of moments of frequency distribution. *Biometrika*, pages 185–210, 1921.

[Utgoff et Clouse, 1996] P.E. Utgoff and J.A. Clouse. A kolmogorov-smirnov metric for decision tree induction. Technical Report TR 96-3, Department of Computer Science - University of Massachusetts, 1996.

[Utgoff, 1989a] P. Utgoff. Incremental induction of decision trees. *Machine Learning*, 4:161–186, 1989.

[Utgoff, 1989b] P. Utgoff. Perceptron trees: A case study in hybrid concept representations. *Connection Science*, 1(4):377–391, 1989.

[Utgoff, 1994] P. Utgoff. An improbved algorithm for incremental induction of decision trees. In *Proceedings of the 11th International Conference on Machine Learning*, pages 402–407. Morgan Kaufmann, 1994.

[Utgoff, 1995] P. Utgoff. Decision tree induction based on efficient tree restructuring. Technical Report 95-18, Department of Computer Science - University of Massachussets, 1995.

[Valiant, 1984] L.G. Valiant. A theory of learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vapnik, 1982] V. Vapnik. *Estimation of dependences based on empirical data*. Springer-Verlag, New York, 1982.

[Ventsel, 1973] H. Ventsel. *Theorie des probabilites*. MIR - Moscou, 1973.

[Venturini, 1994] G. Venturini. *Apprentissage adaptatif et apprentissage supervise par Algorithme Genetique*. PhD thesis, Universite de Paris-Sud, 1994.

[Vilalta *et al.*, 1995] R. Vilalta, G. Blix, and L. Rendell. The value of lookahead feature construction in decision tree induction. Technical Report UIUC-BI-AI-95-01, The University of Illionois at Urbana-Champaign, 1995.

[Volle, 1985] M. Volle. *Analyse des donnees.* Economica, 1985.

[Wald et Wolfowitz, 1940] A. Wald and J. Wolfowitz. On a test whether two samples are from the same population. *Ann of Math. Stat.*, 11:147–162, 1940.

[Wallace et Boulton, 1968] C.S. Wallace and D.M. Boulton. An information measure for classification. *Computer Journal*, 11:185–194, 1968.

[Wallace et Dowe, 1994] C.S. Wallace and D.L. Dowe. Intrinsic classification by mml - the snob program. In *Proceedings of the 7th Australian Joint Conference on Artificial Intelligence*, pages 37–44, Singapore, 1994. World Scientifc.

[Wallace et Freeman, 1987] C. S. Wallace and P. R. Freeman. Estimation and inference by compact encoding (with discussion). *Journal of the Royal Statistical Society series B*, 49:240–265, 1987.

[Wallace et Patrick, 1993] C.S. Wallace and J.D. Patrick. Coding decision trees. *Machine Learning*, 11:7–22, 1993.

[Wang et Suen, 1984] Q.R. Wang and C.Y. Suen. Analysis and design of decision tree based on entropy reduction and its application to large character set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:406–417, 1984.

[Watanabe, 1981] S. Watanabe. Pattern recognition as a quest for minimum entropy. *Pattern recognition*, 13:381–387, 1981.

[Watkins, 1987] C.J. Watkins. Combining cross-validation and search. In *Proceedings of the 2nd European Working Session on Learning*, pages 79–87, 1987.

[Wehenkel et Pavella, 1991] L. Wehenkel and M. Pavella. Decision trees and transient stability of electric power system. *Automatica*, 27(1):115–134, 1991.

[Wehenkel, 1990] L. Wehenkel. *Une approche de l'intelligence artificielle appliquee a l'evaluation de la stabilite transitoire des reseaux electriques.* PhD thesis, Faculte des Sciences Appliquees - Universite de Liege, 1990.

[Wehenkel, 1992] L. Wehenkel. A probabilistic framework for the induction of decision trees. A ete propose mais n'a jamais ete publie sous cette forme. Les principaux resultats ont ete repris dans l'article de 1993., 1992.

[Wehenkel, 1993] L. Wehenkel. Decision tree pruning using an additive information quality measure. In B. Bouchon-Meunier, L. Valverde, and R. Yager, editors, *Uncertainty in Intelligent Systems*, pages 397–411. Elsevier, North Holland, 1993.

[Wehenkel, 1995] L. Wehenkel. Coupling of k-nn with decision trees for power system transient stability assessment. In *Proceedings of IEEE Conference on Control Applications*, 1995.

[Wehenkel, 1996] L. Wehenkel. On uncertainty measures used for decision tree induction. In *Proceedings of Info. Proc. and Manag. Of Uncertainty*, pages 413–418, 1996.

[Wehenkel, 1997] L. Wehenkel. Discretization of continuous attributes for supervised learning - variance evaluation and variance reduction. In *Proceedings of International Fuzzy Systems Association World Congress*, 1997.

[Weiss et Indurkhya, 1993] S. Weiss and N. Indurkhya. Optimized rule induction. *IEEE Expert*, 8(6):61–69, 1993.

[Weiss et Indurkhya, 1994] Sholom M. Weiss and Nitin Indurkhya. Decision tree pruning: Biased or optimal In *Proceedings of the 12th National Conference on Artificial Intelligence. Volume 1*, pages 626–632, Menlo Park, CA, USA, July31 August–4 1994. AAAI Press.

[Weiss et Kulikowski, 1991] S.M. Weiss and C.A. Kulikowski. *Computer Systems that Learn.* Morgan Kaufmann, San Mateo, CA, 1991.

[White et Liu, 1994] A.P. White and W.Z. Liu. Bias in information-based measures in decision tree induction. *Machine Learning*, 15(3):321–329, 1994.

[Wnek et Michalski, 1994] J. Wnek and R.S. Michalski. Hypothesis-driven constructive induction in aq17-hci : A method and experiments. *Machine Learning*, 14:139–168, 1994.

[Wolpert, 1992] D.H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[Wolpert, 1994] D.H. Wolpert. Off-training set error and a priori distinctions between learning algorithms. Technical Report SFI TR 94-12-123, The Santa Fe Institute, 1994.

[Wolpert, 1996] D.H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

[Wong et Chiu, 1987] A.K.C. Wong and D.K.Y. Chiu. Synthesizing statistical knowledge from incomplete mixed mode data. In *IEEE Transaction on Pattern Analysis and Machine Learning*, pages 796–805, 1987.

[Yang *et al.*, 1991a] D. Yang, G. Blix, and L.A. Rendell. A scheme for feature construction and comparison of empirical methods. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 699–704, 1991.

[Yang *et al.*, 1991b] D.S. Yang, L. Rendell, and G. Blix. Fringe-like feature construction : A comparative study and a unifying scheme. In *Proceedings of the 8th International Conference in Machine Learning*, pages 223–227, San Mateo, 1991. Morgan Kaufmann.

[Yip et Webb, 1994] S.P. Yip and G.I. Webb. Incorporating canonical discriminant attributes in classification learning. In *Proceedings of the 10th Canadian Conference on Artificial Intelligence*, pages 63–70. Morgan Kaufmann, 1994.

[Zadeh, 1965] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[Zahn, 1971] C.T. Zahn. Graph theoritical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, C-20(1), 1971.

[Zaki *et al.*, 1996] M.J. Zaki, S. Parthasarathy, W. Li, and M. Ogihara. Evaluation of sampling for data mining of association rules. Technical Report 617, Computer Science Department - The University of Rochester, 1996.

[Zhou et Dillon, 1991] X. Zhou and T.S. Dillon. A statistical-heuristic feature selection criterion for decision tree induction. *IEEE Trans. Pattern Analysis and Machine Learning*, PAMI-13:834–841, 1991.

[Zighed *et al.*, 1992] D.A. Zighed, J.P. Auray, and G. Duru. *Sipina : Methode et logiciel*. Lacassagne, 1992.

[Zighed *et al.*, 1996] D.A. Zighed, R. Rakotomalala, and S. Rabaseda. A discretization method of continuous attributes in induction graphs. In *Proceedings of the 13th European Meetings on Cybernetics and System Research*, pages 997–1002, 1996.

[Zighed *et al.*, 1997] D.A. Zighed, R. Rakotomalala, and F. Feschet. Optimal multiple intervals discretization of continuous attributes for supervised learning. In *Proceedings of the 3rd International Conference in Knowledge Discovery in Databases*, 1997.

[Zighed et Rakotomalala, 1996a] D.A. Zighed and R. Rakotomalala. A method for non arborescent induction graphs. Technical report, Laboratory ERIC, University of Lyon 2, 1996.

[Zighed et Rakotomalala, 1996b] D.A. Zighed and R. Rakotomalala. $SIPINA_W(c) for Windows : User's Guide. Laboratory ERIC - University of Lyon 2, 1996.$

[Zighed, 1985] D.A. Zighed. *Methodes et outils pour les processus d'interrogation non arborescents*. PhD thesis, Universite Claude Bernard - Lyon 1, 1985.